



ESTGOH

**Escola Superior de Tecnologia e Gestão
de Oliveira do Hospital**

Instituto Politécnico de Coimbra

PMS – Quinta dos Mistérios, Turismo de Habitação

**Relatório apresentado com vista à obtenção do grau de Mestre no âmbito da
realização do ciclo de estudos de Mestrado em Informática Aplicada**

Autor:

Eduardo Manuel Furtado Mota

Orientador:

Professor Doutor Luis Veloso

Coorientador:

Professor Nuno Gil

Data: Dezembro de 2016

Dedicatória

Aos meus pais e irmão.

Sem eles, nada disto era possível.

Agradecimentos

À Escola Superior de Tecnologia e Gestão de Oliveira do Hospital, porque durante todos estes anos proporcionou sempre um ambiente amigável e de proximidade com o corpo docente.

A todos os professores da instituição e em especial ao Eng. Luís Veloso e ao Eng. Nuno Gil, orientador e coorientador, respectivamente. Sem eles e sem os seus ensinamentos muito provavelmente não estaria hoje a elaborar esta dissertação. Obrigado por toda a ajuda na realização e culminar da minha vida académica.

A toda a minha família por toda a força que sempre me deu para chegar até este ponto.

À minha mãe que sempre me incentivou e sempre me puxou para cima nos momentos difíceis.

Ao meu pai por todo o apoio e força sem nunca me deixar desmotivar.

A todos os meus amigos, em especial ao Paulo Seixas e ao Carlos Campos, pela boa disposição e companheirismo durante todos estes anos.

Um agradecimento à Fabiana. Ela foi um dos motores principais para a realização deste trabalho sem nunca me deixar perder a esperança e dando-me sempre a força necessária para ultrapassar os inúmeros obstáculos com que me deparei. Obrigado por tudo.

A todos aqueles que, directa ou indirectamente fizeram parte deste trabalho.

Resumo

Actualmente tornou-se quase impensável a existência de uma empresa sem que esta esteja ligada a sistemas de informação. Com estes sistemas as empresas tornam-se mais competitivas, mais eficientes, menos permeáveis a erros e, conseqüentemente, mais produtivas.

O presente trabalho foi desenvolvido no âmbito de uma empresa recente no ramo do turismo que não possuía qualquer sistema de gestão hoteleira, também conhecido por *Property Management Systems* (PMS). Este *software* é imprescindível a qualquer negócio ligado ao turismo, uma vez que é muito importante para este tipo de empresa monitorizar o seu cliente, desde que faz uma reserva até ao momento em que faz o *check-out*. Foi, por isso, desenvolvido neste trabalho um PMS em parceria com a empresa Quinta dos Mistérios - Turismo de Habitação, que o irá utilizar no seu dia-a-dia, fazendo dele a sua principal ferramenta de trabalho.

Após uma avaliação das necessidades da empresa (tendo em conta a inexperiência da mesma) optou-se por uma solução de *software* modular, que facilmente pode ser acrescentado ou moldado ao crescimento da empresa. Esta solução irá criar uma vantagem competitiva trazendo também uma melhor segurança, produtividade e rapidez.

O PMS desenvolvido encontra-se focado no cliente, nomeadamente na facilidade de este realizar as suas reservas. Estas podem ser efectuadas *online*, através de um *website* existente, ou através do PMS, por funcionários trabalhadores da empresa.

O trabalho desenvolvido apresentou, em alguns pontos, uma diferenciação das soluções já existentes no mercado.

Foi possível verificar que com esta aplicação a empresa se tornou mais competitiva, diferenciada e menos permeável a erros, oferecendo um atendimento mais rápido.

Este documento foi elaborado sem respeitar as regras do Novo Acordo Ortográfico.

Palavras-chave

Quinta dos Mistérios, PMS, Turismo de Habitação, Turismo, PHP, *JavaScript*, SQL, CodeIgniter, Bootstrap, HTML, CSS, SQL.

Abstract

Nowadays it is unthinkable to have a company not linked to information systems. With these systems companies become more competitive, more efficient, less permeable to errors, and therefore, more productive.

The study was conducted for a recent tourism company that didn't have any hotel management system also known as *Property Management Systems* (PMS). This type of software is essential to any company linked to the tourism sector, since it is very important for this type of business to monitor all its customers from the moment they make a reservation until they check out. This way, a PMS was developed in the present work in partnership with the Quinta dos Mistérios - Turismo de Habitação Company, which will be used daily, making it their main working tool.

When evaluating the company's needs and knowledge it was decided to implement a modular software solution that can easily be added or molded to the company's growth. This solution will create a competitive advantage, bringing better security, productivity and speed.

The developed PMS focuses on the client, namely in the ease of booking the reservations. Reservations can be made online, through an existing website, and through the PMS (by workers of the company).

The developed work has shown some differentiation regarding the existing solutions on the market.

We were able to show that with this application the company became more competitive, differentiated, less permeable to errors and providing faster service.

This document has been written without following the New Orthographic Agreement.

Keywords

Quinta dos Mistérios, PMS, Rental Accommodation, Tourism, PHP, JavaScript, SQL, Codelgniter, Bootstrap, HTML, CSS, SQL.

Índice

1.	Introdução	1
2.	Estado da Arte.....	3
3.	Calendarização.....	9
3.1.	1º Semestre	9
3.2.	2º Semestre	9
4.	Objectivos e metodologias utilizadas.....	11
5.	Trabalho desenvolvido	13
5.1.	Base de Dados.....	15
5.2.	<i>Front-Office</i>	17
5.3.	<i>Back-Office</i>	19
5.4.	Principais Módulos	24
5.4.1.	<i>Home</i>	24
5.4.2.	Reservas.....	26
5.4.3.	Hóspedes.....	28
5.4.4.	Quartos	30
5.4.5.	Funcionários	32
5.4.6.	Utilizadores	32
5.4.7.	<i>Front-Office</i>	34
5.4.8.	Produtos.....	35
5.4.9.	Pagamentos.....	36
5.4.10.	Calendário.....	37
5.4.11.	Visitantes	38
5.4.12.	Notificações	38
6.	Testes e Resultados.....	41
7.	Conclusões.....	43
8.	Bibliografia.....	47
9.	Anexos	49

9.1.	Anexo I - Lista de Requisitos.....	49
9.2.	Anexo II - Base de Dados	77
9.3.	Anexo III - Manual do Utilizador	83

Lista de Figuras

Figura 1 - Mapa de Gantt, Calendarização Inicial	10
Figura 2 - Mapa de Gantt, Calendarização Final.....	10
Figura 3 - Modelo lógico da Base de Dados	16
Figura 4 - Página <i>Web</i>	18
Figura 5 - <i>Back-Office, Home</i>	19
Figura 6 - Diagrama de estados das Reservas	26
Figura 7 - Aplicação / Informação Clientes.....	29
Figura 8 - Diagrama de actividade - Algoritmo verifica capacidade.....	30
Figura 9 - Modelo ER Base de Dados.....	82
Figura 10 - Adicionar Pré-Reserva - Passo 1	83
Figura 11 - Adicionar Pré-Reserva - Passo 2.....	83
Figura 12 - Adicionar Pré-Reserva - Passo 3.....	84
Figura 13 - Adicionar Pré-Reserva - Passo 4	84
Figura 14 - Adicionar Pré-Reserva - Passo 5.....	85
Figura 15 - Consultar Pré-Reserva – Passos 1 e 2	85
Figura 16 - Confirmar, Apagar ou Consultar Pré-Reserva.....	86
Figura 17 - Confirmar Pré-Reserva – Passo 3	86
Figura 18 - Cancelar Pré-Reserva – Passo 4.....	87
Figura 19 - Consultar ou Cancelar uma Reserva	87
Figura 20 - <i>Check-In</i> - Passo 1	88
Figura 21 - <i>Check-In</i> - Passo 2	88
Figura 22 - <i>Check-In</i> - Passos 3 e 4.....	89
Figura 23 - <i>Check-In</i> - Passos 5 e 6.....	89
Figura 24 - <i>Check-In</i> - Passos 7 e 8.....	90
Figura 25 - <i>Check-In</i> - Passos 9 e 10.....	90
Figura 26 - <i>Check-Out</i> - Passo 1.....	91
Figura 27 - <i>Check-Out</i> - Passo 2.....	91

Figura 28 - <i>Check-Out</i> - Passos 3 e 4.....	92
Figura 29 – Adicionar/Remover Extras - Passo 1.....	92
Figura 30 – Adicionar/Remover Extras - Passos 2 e 3.....	93
Figura 34 - Alterar Dados e <i>Password</i> - Passo 1	93
Figura 35 - Alterar Dados e <i>Password</i> - Passos 2 e 3.....	94
Figura 36 - Adicionar Cliente - Passos 1 e 2	94
Figura 37 - Adicionar Cliente - Passos 3 e 4	95
Figura 38 - Editar e Bloquear/Desbloquear Cliente - Passos 1 e 2.....	95
Figura 39 - Editar e Bloquear/Desbloquear Cliente - Passos 3, 4 e 5.....	96
Figura 40 - Remover Cliente - Passos 1 e 2	96
Figura 41 - Remover Cliente	97
Figura 42 - Remover Cliente - Passo 3	97
Figura 43 - Adicionar Funcionário - Passos 1 e 2.....	97
Figura 44 - Adicionar Funcionário - Passos 3, 4 e 5.....	98
Figura 45 - Editar e Bloquear/Desbloquear Funcionário - Passos 1 e 2	98
Figura 46 - Editar e Bloquear/Desbloquear Funcionário - Passos 3, 4 e 5	99
Figura 47 - Remover Funcionário - Passos 1 e 2.....	99
Figura 48 - Remover Funcionário.....	100
Figura 49 - Remover Funcionário - Passo 3.....	100
Figura 50 - Adicionar Vencimento - Passos 1 e 2	100
Figura 51 - Adicionar Vencimento - Passo 4	101
Figura 52 - Cancelar Vencimento - Passos 1 e 2	101
Figura 53 - Cancelar Vencimento - Passo 3.....	102

Lista de Tabelas

Tabela 1 - Módulo Produtos	54
Tabela 2 - Módulo Funcionários	55
Tabela 3 - Módulo Vencimento Funcionários	56
Tabela 4 - Módulo Quartos	60
Tabela 5 - Módulo <i>Front-Office</i>	62
Tabela 6 - Módulo Notificações	63
Tabela 7 - Módulo Visitantes	64
Tabela 8 - Módulo Pagamentos	64
Tabela 9 - Módulo <i>Home</i>	66
Tabela 10 - Módulo Hóspedes	66
Tabela 11 - Módulo Reservas	72
Tabela 12 - Módulo Calendário	73
Tabela 13 - Módulo Clientes	74
Tabela 14 - Módulo Acompanhantes/Convidados	75
Tabela 15 - Módulo Manual	75
Tabela 16 - Módulo Conta	76
Tabela 17 - Requisitos Não Funcionais	76

Lista de Acrónimos

API - *Application Programming Interface*. Software que contém funcionalidades próprias que não interferem com outros *softwares*.

Controller - Ficheiros PHP, encarregues de todo o funcionamento da aplicação. Estes ficheiros comunicam com todos os outros ficheiros da aplicação (*model*, *view* e *controller*).

CSS - *Cascading Style Sheet*.

Entidade - Representação abstracta de um objecto. Neste relatório são representadas como tabelas de Base de Dados.

HREF - Especifica o URL para onde a página vai.

Icon - Representação gráfica de um programa, ficheiro ou funcionalidade.

IDE - *Integrated Development Environment*. Programa de computador que contém ferramentas que ajudam a desenvolver *software*.

JavaScript - Linguagem de programação executada a partir do interior de um programa por parte do cliente sem ser necessário passar pelo servidor.

Model - Ficheiros PHP, encarregues de tratar de todos os pedidos à base de dados. Comunicam apenas com o controlador.

MVC – Modelo Visão Controlador

PHP – *Hypertext Preprocessor*.

PMS – *Property Management Systems*

PoS - *Point of Sale*. Este ponto de venda tanto pode ser uma máquina registadora, um ponto de venda electrónico ou um local onde exista uma transacção de venda.

Query SQL - Método para retirar dados de uma base de dados e apresentá-los de forma a poderem ser tratados.

View - Ficheiros PHP, encarregues pela parte visual da aplicação. Comunicam apenas com o controlador.

1. Introdução

Com o actual significativo crescimento do turismo em Portugal e o consequente aumento das empresas deste sector, o aparecimento de novos sistemas de gestão hoteleira é inevitável.

Existem ainda muitas unidades hoteleiras, especialmente de pequena escala, que não possuem qualquer sistema informático para ajudar os seus colaboradores a fornecer um serviço mais eficaz e mais rápido aos seus clientes. (1)

Normalmente, estas empresas não sabem com exactidão quantos clientes têm, quantas noites fica cada cliente, ou até mesmo a probabilidade de voltarem ou não.

Neste sector o cliente tem de ser tratado como um indivíduo único, devendo as empresas guardar dados específicos sobre os mesmos. Desta forma, é possível realizar um atendimento mais personalizado, valorizando cada cliente como pessoa e não apenas como um número.

Existem no mercado muitos sistemas de gestão hoteleira. Contudo, estes sistemas não se encontram habitualmente escalados para o negócio que muitas casas de turismo ou pequenas pousadas exigem. Normalmente são empreendimentos de pequena escala, que necessitam de sistemas mais focados nas ofertas das suas unidades hoteleiras, e não tanto na parte financeira do negócio. (2)

Muitos dos sistemas existentes no mercado não dão uma real ênfase ao cliente, desde que este faz uma pré-reserva até ao momento em que faz o *check-out*. Este tipo de turismo requer um contacto próximo com os seus clientes e, por isso, é imperativo um sistema que proporcione essa proximidade.

Como jovem empreendedor e sem experiência na área de hotelaria, resolvi desenvolver a minha própria aplicação de gestão hoteleira para colmatar aquilo que me parece faltar em todas as aplicações por mim analisadas.

A falta de experiência deveria funcionar como um alerta para que eu fizesse exactamente o oposto do que fiz. Supostamente, eu deveria comprar um sistema gestor já existente no mercado, em que muitos dos problemas já estão resolvidos, e pronto a utilizar. Mas cedo abandonei esta ideia, pois com o meu próprio sistema gestor posso adicionar ou retirar módulos à medida que as dificuldades vão surgindo, o que me poderá permitir ter uma vantagem competitiva em relação aos outros utilizadores deste tipo de *software*.

Esta foi a principal motivação para avançar com este projecto, o qual permitirá incrementar a minha aplicação com novos e melhores módulos, que se adequam 100% ao meu negócio.

Outro factor que levou à construção da aplicação foi a questão financeira e a constatação de que, apesar de existirem no mercado muitas ofertas livres, nenhuma ia ao encontro das necessidades desta pequena empresa. Muitas por serem demasiado complexas, outras por serem demasiado simples. As soluções encontradas que mais se adequavam às necessidades da empresa eram *software* pago.

A aplicação foi desenvolvida com o intuito de ser utilizada diariamente pelo promotor e seus colaboradores. A aplicação não conta com um módulo financeiro complexo, uma vez que a empresa terá de usar um programa homologado para este fim. Em contrapartida, é focada nos módulos do *Front-Office* (página *web*), nas reservas, nos quartos e finalmente nos clientes, as peças centrais de qualquer negócio de hotelaria.

Toda a aplicação foi desenvolvida para ir ao encontro dos parâmetros estabelecidos pela empresa Quinta dos Mistérios - Turismo de Habitação, com o intuito de criar uma aplicação funcional, confiável, eficiente e acessível em qualquer local. Adicionalmente, pretende-se que seja de fácil utilização, com a possibilidade de ser adaptável a novas necessidades que possam surgir.

A referida empresa consiste num negócio familiar de turismo que se localiza na ilha de S. Jorge, no Arquipélago dos Açores.

2. Estado da Arte

Actualmente existem no mercado várias soluções de sistemas gestores de hotelaria. Estes sistemas podem ser pagos ou de licença gratuita, para negócios da pequena à grande escala, mais focados na parte financeira ou no cliente, entre outras opções, havendo uma ampla oferta para as mais variadas escolhas.

Dada a diversidade existente, foi necessário estabelecer critérios relativamente às aplicações que seriam escolhidas para análise. Os principais critérios foram a cotação que estas aplicações possuem (feitas pelos utilizadores), o tamanho do negócio a que se destinam, a contemporaneidade e se são sistemas acessíveis num ou em vários terminais.

A escolha das aplicações para analisar foi baseada no *Capterra Hotel Management Blog*, tendo sido seleccionados dois critérios fundamentais. O primeiro refere-se às aplicações livres mais votadas por parte dos utilizadores deste tipo de *software* (3) e o segundo às melhores aplicações para negócios de pequena escala (4). De salientar que apenas puderam ser testadas as aplicações pagas que tinham a versão experimental.

A análise das aplicações foi feita através da descrição do funcionamento dos principais módulos e, muitas vezes, com uma explicação de como efectuar determinada acção. Adoptou-se este método para, desta forma, alcançar um maior detalhe da aplicação e verificar se o funcionamento da mesma é o mais correcto ou mais prático.

Com uma descrição das aplicações mais exaustiva, tornou-se mais simples detectar as principais vantagens e desvantagens em relação ao trabalho desenvolvido.

Foram seleccionadas 7 aplicações para serem analisadas, sendo escolhido como critério principal o tamanho do negócio a que se destinam.

Desta forma, foi analisado 1 sistema gestor direccionado para negócios de grande escala, 1 para negócios de grande a média escala e 5 sistemas gestores para negócios de média a pequena escala. Foi dada especial atenção a estes últimos, visto serem da mesma escala que a empresa Quinta dos Mistérios, Turismo de Habitação.

Abacre Cloud Hotel Management System (5) - Aplicação para negócios de grande escala que funciona em ambiente Windows, tendo a base de dados ligada a uma *cloud*, fornecida pelo desenvolvedor deste *software*.

No topo da página da aplicação existem 7 menus e cada um possui diferentes funcionalidades:

- No menu “acções” podemos escolher as opções: “*novo pedido*”, “*imprimir conta*”, “*fechar pedido*”, “*dividir pedido*”, “*log off*”, “*iniciar caixa*”, “*acertar caixa*”, “*fim do dia*” e “*sair*”.
- O menu “vistas”, que tem como opções: “*pedidos*”, “*cliente*”, “*turnos*”, “*relatórios*”, “*email sender*”, “*barra de ferramentas*”, “*linguagem*” e “*temas*”.
- O menu “hotel” tem como opções: “*reservas*”, “*estadias*”, “*quartos*”, “*tipo de quartos*”, “*calendário dos quartos*”, “*cálculo dos custos diários dos quartos*”.
- O menu “inventário”, apresenta como opções: “*inventário físico*”, “*categorias do inventário*” e “*unidades de inventário*”.
- No menu “compras” as opções são: “*comprar encomenda*”, “*receber vouchers*”, “*devolver voucher*”, “*vendedores*” e “*fornecedores*”.
- No menu “configurações” as opções são: “*configurações do hotel*”, “*preferências funcionários*”, “*tipos de funcionários*”, “*produtos do menu*”, “*categorias dos itens de menu*”, “*modificações dos produtos do menu*”, “*mesas*”, “*plano das mesas*”, “*configuração das contas*”, “*métodos de pagamento*”, “*notificações da impressora*”, “*tipo de pedidos*”, “*tipo de clientes*” e “*descontos*”.
- O último menu é o “Help”. Este tem como opções: “*conteúdos*”, “*fórum*”, “*feedback*”, “*gostar deste produto no facebook*”, “*ver comércio do Abracre*” e “*sobre*”.

Como seria de esperar, esta aplicação destina-se a um negócio de grande escala sendo por isso muito focado na gestão de stocks e produtos, em detrimento dos hóspedes ou clientes.

Esta aplicação é direccionada para um hotel com restaurante, sendo este o módulo melhor desenvolvido, tendo sempre activos vários botões de atalhos para os pedidos.

Dentro das desvantagens encontra-se a lentidão que muitas vezes é detectada em acções simples, como realizar uma mera pesquisa. Também não é muito claro o sistema de funcionamento das reservas ou dos clientes. O utilizador tem de saltar constantemente de página em página para efectuar uma acção simples, como uma reserva.

A aplicação é complicada de utilizar, encontrando-se em falta um bom manual de utilizador.

As grandes vantagens são os módulos de gestão de produtos e a possibilidade de alterarmos as configurações de acordo com a preferência do utilizador. O módulo dos produtos revela ser muito completo, com várias opções para a gestão de *stocks*, adicionar fornecedores, fazer encomendas, entre outros.

HotelRunner (6) – Sistema gestor para negócios de grande a média escala. Este *software*, metade livre e metade pago, é utilizado por grandes cadeias de hotéis, como a cadeia IBIS ou Sheraton. Trata-se de uma aplicação totalmente *online*, com funcionamento em *cloud*.

Ao analisarmos esta aplicação ficamos sem saber o verdadeiro potencial da mesma, pois não tivemos acesso à versão paga. A aplicação apresentava um *layout* simples que aparentava ser de fácil utilização, o que nem sempre se confirmou. No final vemos uma aplicação um pouco básica. Faltam, claramente, mais módulos e abordagens mais fáceis para tarefas menos complexas.

Podemos salientar como pontos fortes a possibilidade de ter um *website* totalmente gratuito, que pode ser editado através do *Back-Office*, e a integração da aplicação com outros canais de distribuição como o TripAdvisor, Booking ou outras agências de viagens online.

Como pontos fracos apontamos os módulos “*reservas*”, “*quartos*” e “*clientes*”. Desconhecemos se estas funcionalidades se encontram completas mas, na versão ensaiada, não é muito intuitivo o modo de adicionar novos quartos e de torná-los disponíveis para reservar. A informação sobre os clientes é pouca e não existe a possibilidade de alterar os estados de uma reserva, como *check-in* ou *check-out*. O estado das reservas apenas está confirmado ou por confirmar, não existindo no sistema o conceito de cliente hospedado.

Na construção do *website* não é possível publicar serviços ou promoções, nem realizar uma descrição um pouco mais pessoal dos quartos, ou mesmo da pousada ou hotel.

Outro ponto negativo é a inexistência da possibilidade de apagar a conta criada no HotelRunner.

Base7booking (7) - Aplicação *online* de licença paga para negócios de média a pequena escala.

Este *software* mostrou ser bastante simples de usar, com um *layout* de fácil manuseamento e compreensão.

Os pontos fortes são as opções “*contabilidade*” e “*estatísticas*”, que mostraram ser muito completas e com informação rica para uma utilização real do sistema.

Outra vantagem da aplicação, e peça central da mesma, é o módulo “*calendário*”. Este módulo tem várias funcionalidades como filtros de pesquisa por dias, por quarto e por tipo de quarto. Também possibilita a marcação reservas através do calendário e, entre outras opções, a verificação do estado do quarto (e.g. se está sujo ou limpo).

As grandes desvantagens começam pelo uso de um *Front-Office* distinto para fazer as reservas. Não é possível (na versão ensaiada) efectuar uma reserva através do *Back-Office*. Adicionalmente, o *Front-Office* guarda os dados do cliente anterior, fazendo com que o utilizador tenha de apagar todos os campos para reintroduzir um novo cliente.

A não existência de um *website*, também é um ponto negativo. O *Front-Office* consiste apenas num sistema de reservas por passos e não num *website* que ajude a vender o negócio.

Na componente do *Back-Office*, não é explícito como funciona o pagamento das reservas / estadias, nem como estas mudam de estado. Por exemplo, ao carregar no botão *check-in* a reserva não actualiza os seus dados. Outra desvantagem é a falta de informação sobre os convidados (acompanhantes) dos clientes.

eZeeFrontDesk (8) – Esta aplicação paga para sistemas de média a pequena escala oferece três módulos distintos. Um módulo “*Front-Office*”, um módulo “*Back-Office*” e um módulo “*PoS*”.

Como se trata de um sistema bastante completo, a sua utilização torna-se complexa, não sendo muito explícito o modo de fazer um *check-in* ou *check-out*.

Como pontos fortes assinala-se o facto de ser uma aplicação muito completa, tanto a nível de dados dos clientes como dados das reservas, apresentando ainda uma opção para a restauração.

O módulo “*PoS*” incorporado é sem dúvida uma grande vantagem, sendo poucas as aplicações que trazem este tipo de módulo já incorporado.

Como pontos fracos assinalamos a necessidade do uso dos três módulos em simultâneo para o correcto funcionamento da aplicação.

Não é um *software* de fácil manuseamento e compreensão. As mais de 100 opções de menus que apresenta tornam-na excessivamente grande para hotéis de pequenas dimensões ou para casas de turismo de habitação.

RoomMater Standard Edition (9) - Esta aplicação para hotéis de média a pequena escala está focada no sistema de reservas, sendo este bastante completo.

O *layout* do *software* mostra ser antiquado e nem sempre é fácil fazer uma acção básica, como uma nova reserva. O funcionamento da disponibilidade dos quartos não é de fácil compreensão e, talvez por se tratar da versão experimental, não existe a opção de adicionar um novo quarto ou de editar as datas (época alta e baixa) de um quarto já existente.

Como aspectos negativos salientamos a quantidade excessiva de informação que muitas vezes aparece no ecrã. Por exemplo, ao carregarmos na informação de uma reserva temos mais de 20 botões com novas informações e um formulário com mais de 25 caixas de texto. Outro ponto negativo é a falta de importância dada ao cliente, não existindo uma opção para listar todos os clientes.

Como pontos fortes salientamos um conjunto muito completo de relatórios de todas as actividades/acções do hotel. Outro ponto positivo desta aplicação é a integração de um sistema PoS para a venda de produtos ou para a restauração.

HotelOgix (10) – Este sistema gestor pago mostrou ser simples e, por isso, bastante funcional. É uma aplicação direccionada a hotéis de média a pequena escala, focada nas reservas, nos quartos e nos produtos, tendo ainda uma opção extra, que é o restaurante.

Como pontos negativos apontamos o fraco *layout* e a falta de clareza no funcionamento da opção produtos. Não é explícito o modo de associar um produto a uma estadia ou fazer uma venda.

A navegação dentro da aplicação não é a melhor. Sempre que se carrega num *link* do menu, um novo separador no *browser* é aberto e somos direccionados para lá, dando a sensação que não se está dentro da mesma aplicação.

Como pontos fortes apontamos a simplicidade que, salvo uma ou outra funcionalidade, faz com que a aplicação seja de fácil utilização e bastante intuitiva. A opção de compra de diferentes pacotes para diversos tipos de negócio (*B&Bs*, *City Hotels*, *Resort*, *Serviced Apartments*) optimiza a aplicação, adequando-a às necessidades do comprador.

Outro ponto forte é o calendário bastante funcional, no qual é possível realizar a maioria dos eventos associados ao módulo reservas.

Hoteliga (11) - De todas os sistemas gestores analisados, o Hoteliga é o mais semelhante à aplicação desenvolvida. Sendo desenvolvido através da *framework* Bootstrap, tem como pontos fortes a facilidade de manuseamento e o módulo reservas.

Nesta aplicação é muito fácil adicionar novas reservas, quartos e clientes. Outros pontos fortes são o seu calendário, no qual é possível fazer reservas directamente e o módulo “*relatórios*”, onde é possível exportar todas as folhas para PDF. O módulo “*ferramentas*” tem muitas opções de edição sobre a facturação, a política de preços, os serviços, a limpeza e outros.

Um dos pontos negativos deste *software* é trabalhar sobre janelas modais. Ao utilizarmos esta aplicação num *tablet* de pequenas dimensões torna-se difícil encontrar os botões certos. Outro ponto fraco é o facto de as páginas demorarem muito tempo a carregar. A inexistência de um módulo “*extras*” e a falta de informação devido à simplicidade da aplicação, também são vistos como factores negativos.

Dada a facilidade em inserir um cliente, também seria de esperar a possibilidade de inserir os hóspedes que acompanham os clientes. Contudo, tal não se verifica, não existindo a possibilidade de registar os acompanhantes.

Apenas é possível reservar um quarto por cliente, não havendo a possibilidade de fazer uma reserva para um grupo.

3. Calendarização

3.1. 1º Semestre

- Análise do projecto.
- Criação do repositório de *software*.
- Identificação dos documentos a criar.
- Análise de requisitos.
- Especificação de requisitos.
- Validação de requisitos.
- Protótipos.
- Criação da Base de Dados e modelo ER.

3.2. 2º Semestre

- Implementação da Base de Dados.
- Escolha dos *templates*.
- Implementação de todos os módulos.
- Testes aos módulos implementados.
- Elaboração do relatório final.
- Apresentação e defesa final da dissertação.

A calendarização do projecto não pôde ser cumprida como inicialmente estava previsto. Sendo o empreendedor de todo o projecto (casa de turismo de habitação), tive de iniciar as diligências para concorrer a fundos europeus tendo em vista a realização de todo o projecto Quinta dos Mistérios – Turismo de Habitação.

No início do mês de Março, para obter direito aos fundos, foi necessário realizar um curso de empreendedorismo com uma duração completa de 6 meses. Este curso era dividido em dois percursos de 3 meses cada um. O primeiro percurso foi de Março a Junho de 2015 e o segundo de Setembro a Dezembro do mesmo ano.

Pelas referidas razões, foi pedido um prolongamento de 6 meses para a conclusão da dissertação do mestrado.

Na Figura 1 vemos a calendarização proposta inicialmente.

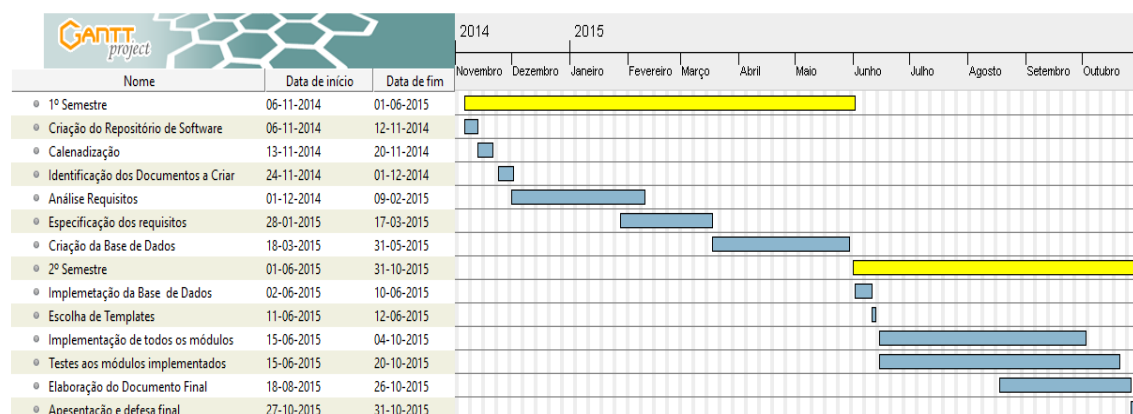


Figura 1 - Mapa de Gantt, Calendarização Inicial

Na Figura 2 vemos a calendarização final.

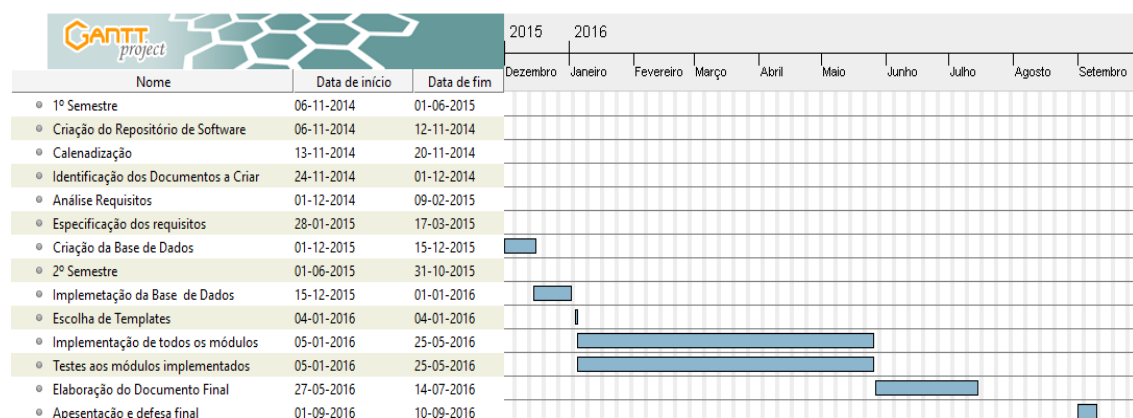


Figura 2 - Mapa de Gantt, Calendarização Final

4. Objectivos e metodologias utilizadas

Era imperativo logo de início criar-se um *website* que fosse a imagem da empresa. A própria aplicação teria de ser de fácil manuseamento e intuitiva.

O principal objectivo foi dar ao promotor da empresa Quinta dos Mistérios – Turismo de Habitação um sistema gestor, de acesso e utilização fáceis, que fosse a sua principal ferramenta de trabalho. Era importante a aplicação ser *web-based* e que fosse acessível em qualquer local, pelos principais *browsers* de *internet*.

A plataforma a desenvolver deveria responder a objectivos como:

- Facilidade de utilização por parte dos utilizadores.
- Ser acessível em qualquer local.
- Os clientes poderem fazer reservas *online*.
- Poder ser usada no dia-a-dia como ferramenta de trabalho.
- Adaptabilidade elevada à introdução de novos módulos.

Para o desenvolvimento da plataforma escolheu-se uma metodologia em cascata. Este modelo é caracterizado por só permitir o avanço para a próxima fase, depois da anterior estar completa.

O modelo escolhido foi dividido em 5 fases:

- 1- Definição de requisitos.
- 2- Projecto do sistema.
- 3- Implementação do código.
- 4- Testes ao código implementado.
- 5- Correção de erros e manutenção do *software*.

Sempre que novos módulos surgiam, o modelo em cascata voltava ao início e novos módulos eram implementados.

Este tipo de modelo foi escolhido pois era importante para o desenvolvedor do projecto ter uma ideia global do mesmo, e de como interagiriam os diferentes módulos que iriam ser implementados.

Outra razão para a escolha desta metodologia foi a possibilidade de identificação dos principais módulos a serem implementados, permitindo assim um maior foco de trabalho e de tempo nestes módulos principais.

A aplicação encontra-se dividida em 15 módulos principais. Entenda-se que um módulo é um conjunto de soluções de programação que fazem funcionar uma parte do programa e que são independentes entre si. Os módulos que se resolveram instalar para o promotor usar a plataforma como ferramenta de trabalho foram:

- *Home.*
- *Hóspedes.*
- *Reservas.*
- *Calendário.*
- *Produtos.*
- *Acompanhantes.*
- *Funcionários.*
- *Quartos.*
- *Manual.*
- *Front-Office.*
- *Notificações.*
- *Check-In.*
- *Check-Out.*
- *Visitantes.*
- *Conta.*

O último objectivo, e talvez o mais ambicioso, é a possível comercialização da aplicação. Para isto ser possível a aplicação não poderá ser apenas centrada na empresa Quinta dos Mistérios. Ou seja, o desenvolvimento da plataforma tem de ser moldável a outras empresas do mesmo ramo de turismo.

5. Trabalho desenvolvido

Para ir ao encontro dos objectivos propostos, a aplicação foi dividida em duas vertentes: o **Front-Office** e o **Back-Office**.

O **Front-Office** consiste numa página *web* que tem como principal funcionalidade possibilitar ao cliente a realização de uma pré-reserva, escolhendo o quarto que deseja, bem como o número de noites. A página principal tem também como funcionalidades uma secção de galeria de fotografias, de promoções, de serviços, de actividades e de contactos. Todas as funcionalidades que tenham alterações mais recorrentes (e.g. promoções) são alteradas através do **Back-Office**.

O **Back-Office** é a parte do sistema em que se procede à gestão da pousada ou do pequeno hotel. Encontra-se dividido em 15 módulos principais, estando acessível apenas a utilizadores registados na aplicação. Os visitantes não têm acesso a esta parte da aplicação e, muitas vezes, nem sabem que ela existe.

Para alojar a aplicação foi alugado um servidor Linux, através de uma empresa prestadora deste tipo de serviços.

Para desenvolver este trabalho foram utilizadas as seguintes tecnologias:

- HTML.
- PHP.
- *JavaScript*.
- *Mysql*.
- *JQuery*.
- CSS.

Adicionalmente foram também utilizadas *frameworks*, APIs e *plug-ins* para implementar as funcionalidades propostas, que se encontram a seguir descritas.

Frameworks:

Bootstrap (12): Esta *framework* foi utilizada para desenhar a aplicação e o *website*. Tem como grande particularidade o facto de possuir um conjunto de funcionalidades que permite adaptar os *layouts* aos diferentes tamanhos de tela. Deste modo é óptimo para visualização, tanto em monitores como em *tablets* ou *smartphones*. Esta *framework* livre e de código aberto contém um conjunto de ficheiros CSS que permite aos programadores implementarem diferentes *toolkits* da *framework*. Estes

componentes de interface, para além de utilizarem HTML e CSS, também utilizam ficheiros do tipo *JavaScript*, aumentando assim a funcionalidade de interacção com os utilizadores deste tipo de *frameworks*.

CodeIgniter (13): esta *framework* foi utilizada para desenvolver o código PHP e funciona através da arquitectura de *software Model View Controller* (MVC). Este diferente paradigma de programação em PHP funciona através de um ficheiro central (*controller*) que é responsável por fazer a ligação dos ficheiros da base de dados (*model*) aos ficheiros HTML (*view*). Esta *framework* livre, de código aberto, orientada a objectos, utiliza um conjunto de bibliotecas internas que formam o *toolkit* CodeIgniter. Este *software* tem como principal objectivo aumentar a produtividade dos seus utilizadores. A capacidade de prevenir ataques a *websites*, a facilidade de instalação, um vasto leque de opções de documentação e a optimização de funcionalidades de programação foram as principais razões pelas quais foi escolhida a *framework* CodeIgniter.

No caso desta aplicação, a denominação dada aos ficheiros do tipo *controller* e *view* no *Back-Office* da Administração foi: admin_nome_do_ficheiro. No *Back-Office* dos Funcionários, a denominação escolhida foi: func_nome_do_ficheiro. No *Front-Office* ambos os ficheiros de tipo *controller* e *view* foram designados nome_do_ficheiro. Os ficheiros do tipo *model* foram denominados m_nome_do_ficheiro.

APIs

ZingChart (14): Esta API encontra-se incorporada na aplicação para controlar e mostrar alguns dados estatísticos sobre os quartos e as estadias. Uma tabela de base de dados foi criada especificamente para trabalhar com esta API.

FullCalendar (15): É através desta API que são mostrados os quartos e os dias em que estes se encontram ocupados. Esta API também possui uma tabela de base de dados específica.

GoogleMaps: No âmbito desta API é possível mostrar a localização exacta onde se encontra a casa da Quinta dos Mistérios. Esta é apenas mostrada na parte do *Front-Office* da aplicação.

Plug-ins

DataTables: Todas as tabelas da aplicação usam o *plug-in datatable*. Este *plug-in* facilita a visualização dos dados. Para além de fazer a paginação automática,

mostrando desde 10 a 100 registos por página, também contém uma ferramenta de procura por *string*, caracter ou numérica.

Para programar a aplicação foi utilizado o IDE *Netbeans* que se encontrava directamente ligado ao servidor, sendo todas as alterações automaticamente enviadas e actualizadas.

A aplicação é guardada num repositório que contém 35 controladores diferentes, 21 ficheiros do tipo *model* e mais de 160 ficheiros do tipo *view*, para além de outros ficheiros do tipo CSS, *JavaScript* e pastas próprias da *framework* CodeIgniter.

5.1. Base de Dados

A base de dados é a componente fundamental e mais importante de qualquer sistema de informação, pois é nela que são guardados os dados relativos aos nossos clientes, quartos, reservas, entre outros.

A presente base de dados relacional é composta por 27 entidades ou tabelas que se encontram relacionadas entre si através de um identificador comum, as chaves primárias.

Para desenvolver a base de dados foram utilizados dois *softwares*:

Power designer- Este *software* contém um conjunto de estruturas e especificações com a representação de um ou mais processos de desenvolvimento de um sistema informático. Este *software* pago foi utilizado para criar a base de dados de entidades e relações.

HeidiSQL – Este *software* livre, de código aberto, utiliza o *MySQL server* e foi utilizado principalmente para fazer pesquisas, pequenas alterações à base de dados e testar *queries* SQL.

A base de dados está guardada num servidor externo que tem como nome **quintad8_turis**. (Figura 3)

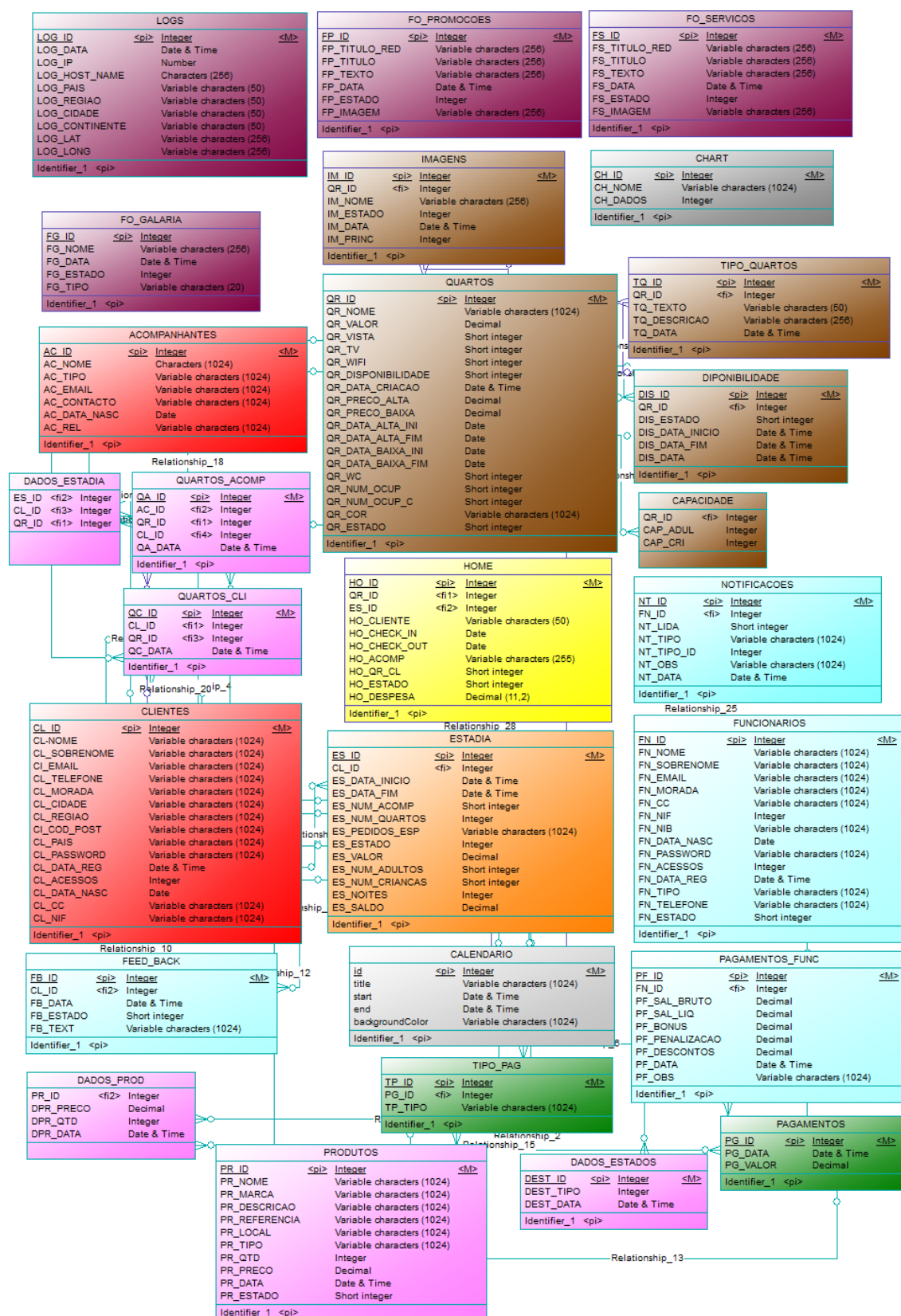


Figura 3 - Modelo lógico da Base de Dados

5.2. *Front-Office*

O *Front-Office* encontra-se direccionado para as reservas efectuadas pelos clientes e para a divulgação da Casa da Quinta dos Mistérios.

A página *web* (Figura 4) foi dividida em três partes principais: o menu de navegação em cima, o corpo da página com todos os conteúdos necessários para o cliente realizar uma reserva e, em baixo, um rodapé com os contactos e localização do empreendimento.

Em cima, o menu tem como opções “*HOME*”, “*GALERIA*”, “*CONTACTOS*”, “*RESERVAS*” e “*LOGIN*”.

A opção “*HOME*” destaca-se pelo calendário no qual o cliente pode iniciar uma pré-reserva. Para isso, apenas terá de escolher os dias de *check-in* e *check-out*, o número de adultos, o número de crianças e carregar no botão “*Reservar Já*”.

Por baixo do calendário existem *hyperlinks* para mostrar os quartos, os serviços, as promoções e a localização, que estão activos para dar a conhecer ao cliente o espaço Casa da Quinta dos Mistérios.

O sistema de reservas por parte do cliente é feito por passos. Após o cliente carregar no botão “*Reservar Já*” é reencaminhado para outra página *web*, na qual todos os quartos são mostrados e onde poderá escolher o quarto que mais lhe agrada, caso este esteja disponível. No passo seguinte, um formulário com os dados do cliente é mostrado e, após o preenchimento do mesmo, avança para o passo final. Este passo final consiste num resumo de todas as escolhas anteriores e apresenta o valor total da estadia. Ao carregar no botão “*Submeter Reserva*” esta é automaticamente guardada no sistema.

A opção “*GALERIA*” mostra um conjunto de fotografias da Casa de Turismo Quinta dos Mistérios. As fotografias estão divididas em quatro categorias: *casa*, *quartos*, *salas* e *actividades*. Através de um filtro, o utilizador pode escolher quais as categorias que deseja ver isoladamente.

Na opção “*CONTACTOS*”, para além da morada e dos números de telefone, o cliente poderá visualizar a localização da casa através do Google Maps. À esquerda existe um pequeno formulário com os campos *Chegada*, *Partida*, número de *Adultos* e número de *Crianças*, para que o cliente possa iniciar uma reserva sem ter de voltar à página inicial (*home*).

Na opção “*RESERVAS*” é mostrado um calendário com a data de chegada e partida, assim como o número de hóspedes e um botão “*Reservar Já*”. A única distinção deste calendário para o da página *home* é que este mostra os três meses seguintes.

A última opção é “*LOGIN*”, através da qual os funcionários e administradores entram no *Back-Office* e na aplicação propriamente dita. Esta opção irá ser retirada no futuro, uma vez que não faz sentido os clientes terem uma opção da qual não podem usufruir.

Toda a página *web* foi elaborada com os tons de salmão e *bordeaux*. A escolha destas cores deveu-se ao facto da existência de um logotipo, já pré-concebido, e que irá ser uma das imagens da marca Casa da Quinta dos Mistérios.



Quinta dos Mistérios

HOME GALERIA CONTACTOS RESERVAS LOGIN

FAÇA A SUA RESERVA

Chegada: 6 Jul, 3:00 PM Partida: 7 Jul, 12:00 PM

Adultos: 1 Crianças: 0

Reservar já!

QUARTOS
Escolha o quarto que deseja. Temos várias à sua escolha.

SERVIÇOS
Veja a nossa piscina e outros serviços.

PROMOÇÕES
Fique cinco noites e ganhe uma completamente **gratuita!**

LOCALIZAÇÃO
Estamos a um minuto do aeroporto.

Figura 4 - Página Web

5.3. Back-Office

O *Back-Office* encontra-se dividido em duas componentes. A referente aos Administradores e a destinada aos Funcionários. Apenas será descrita a parte dos administradores, pois é a mais completa consistindo a diferença para a parte dos funcionários em apenas um maior número de funcionalidades. Os restantes elementos (*layout*, tipo de botões e funcionalidades conjuntas) são iguais em ambos os módulos.

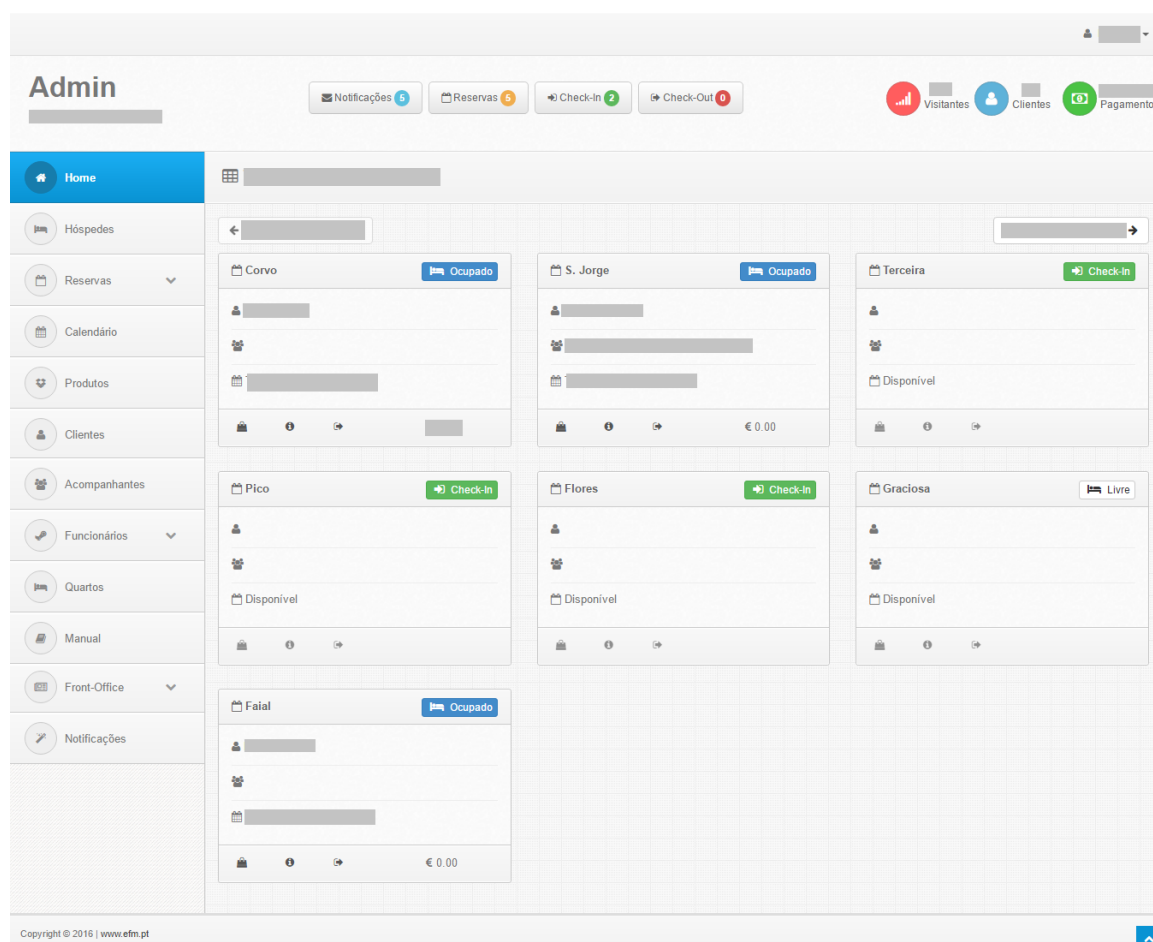


Figura 5 - Back-Office, Home

Ao entrarmos na aplicação, em cima existem quatro botões: “Notificações”, “Reservas”, “Check-In” e “Check-Out”. Estes encontram-se sempre activos e o utilizador pode, em qualquer altura, carregar neles para ver a informação que contêm. Em cima, à esquerda, existe um relógio e do lado direito *links* para as opções “Visitantes”, “Clientes” e “Pagamentos”.

Do lado esquerdo existe um menu sempre activo com as seguintes opções: “Home”, “Hóspedes”, “Reservas”, “Calendário”, “Produtos”, “Clientes”, “Acompanhantes”, “Funcionários”, “Quartos”, “Manual”, “Front-Office” e “Notificações”. (Figura 5)

Na página principal “*Home*”, poderemos visualizar todos os quartos do empreendimento e os respectivos estados no próprio dia. Quando um quarto se encontra ocupado é mostrado o nome do cliente que o ocupa, os acompanhantes/convidados (se existirem), as datas de entrada e saída e, ainda, os botões para adicionar um extra, visualizar a informação da estadia e fazer *check-out*.

Por cima da informação dos quartos existe um botão com a data do dia seguinte. Este botão foi implementado para facilitar a visualização dos eventos de cada quarto nos dias seguintes. Esta abordagem permite mostrar todos os estados dos quartos em função dos dias.

Avançando para a *view* “*Hóspedes*”, uma tabela com os hóspedes que se encontram actualmente na pousada é mostrada. Esta visualização é realizada através de uma tabela, sendo os clientes agrupados juntamente com os seus acompanhantes. Os dados mostrados são: o número do cliente, se é cliente ou acompanhante, o nome, o quarto que ocupa, o *email*, o contacto e a data de saída.

Na opção “*Reservas*” existem duas possibilidades: “*Adicionar*” uma reserva ou visualizar o “*Histórico*” de todas as reservas. Ao adicionarmos uma reserva, um sistema por passos é mostrado ao utilizador.

No primeiro passo, o utilizador terá de escolher um cliente da sua lista de clientes. Se for um novo cliente, o utilizador necessitará de criar primeiro o novo registo e só depois é que poderá escolher o cliente da lista.

No passo seguinte (passo dois), é solicitado ao utilizador a introdução dos dados da reserva: data de chegada, data de saída, número de adultos, número de crianças, se pretende *transfers* e, finalmente, pedidos especiais.

No passo três, são mostrados todos os quartos do empreendimento e o utilizador apenas terá de escolher o quarto disponível que pretende reservar.

No quarto e último passo, um resumo da reserva é mostrado e o utilizador apenas terá de verificar se está em conformidade e carregar no botão “*Adicionar Pré-Reserva*”.

Se for escolhida a opção “*Histórico*”, uma lista com todas as reservas é mostrada. Esta lista, em forma de tabela, tem como campos: data de chegada, data de partida, estado, número de quartos, data de criação, total de hóspedes, nome do cliente e um botão para mostrar toda a informação da reserva.

Ao carregar no botão “*informação*”, uma página tipo folha de pagamento é mostrada com toda a informação da estadia (o custo, o nome dos hóspedes e os produtos consumidos).

A opção seguinte é “*Calendário*”. Aqui, um calendário mensal é mostrado com as reservas dos quartos. Cada quarto tem a sua cor, sendo no calendário mostrada uma barra da cor do quarto, que começa e acaba nos dias em que o mesmo se encontra reservado. Este calendário apenas mostra informação, não sendo possível a sua edição.

Os “*Produtos*” encontram-se todos listados numa tabela. À direita, na tabela, existem três botões: “*informação*”, “*editar*” e “*apagar*”. Por cima da tabela existe um botão “*Adicionar Produto*”.

Para adicionar um produto o utilizador apenas tem preencher um formulário com os dados do produto e carregar no botão “*Adicionar*”. Os campos a preencher são: modelo/nome, marca, referência, tipo, descrição, preço, quantidade e local em armazém.

Ao carregar no botão “*informação*” é mostrada uma tabela com o número da estadia em que o produto foi vendido, a data da venda, a quantidade e o preço unitário. Por cima da tabela temos um conjunto de dados estatísticos, como as quantidades do produto vendidas e o valor do total de vendas.

O botão “*editar*” serve para alterar os dados do produto e tem os mesmos campos que o formulário de adicionar um produto, com a excepção de um botão que serve para activar ou desactivar um produto.

O botão “*apagar*” elimina um produto do sistema.

Na opção seguinte temos os “*Clientes*”. À semelhança de “*Produtos*”, uma lista em forma de tabela é mostrada com todos os clientes e, à direita, na tabela, existem os botões “*informação*”, “*editar*” e “*apagar*”. Por cima da tabela surgirá o botão “*Adicionar Cliente*”. Apenas se descreverá o botão “*informação*”, uma vez que os outros botões têm as mesmas funcionalidades que os de “*Produtos*”.

Ao carregar no botão “*informação*”, é mostrada uma lista com todas as reservas efectuadas pelo cliente. Os campos mostrados são: número da reserva, data de entrada, data de saída, quarto, quantidade de quartos alugados, número de noites, estado da reserva, valor da reserva e gasto em extras. Por cima, temos dados

estatísticos referentes ao cliente como o total de estadias no hotel, a média de noites por estadia e o gasto total.

A opção “*Acompanhantes*” é semelhante a “*Clientes*”. Contudo, nesta opção não existe a possibilidade de apagar um acompanhante, nem existe o botão adicionar acompanhante (estes só podem ser adicionados no *check-in*). Tudo o resto é igual à opção “*Clientes*”.

Avançando para a opção “*Funcionários*”, verifica-se que esta se encontra dividida em duas vertentes. Uma referente aos dados dos funcionários (“*Lista*”) e outra aos vencimentos dos funcionários (“*Vencimentos*”).

Apenas iremos analisar o botão “*informação*”, pelas razões referidas anteriormente. Ao carregarmos no botão “*informação*”, surgem os dados estatísticos do funcionário escolhido, quantos acessos ele fez na aplicação, quantas reservas confirmou, quantos *check-ins* e *check-outs* realizou, número de produtos vendidos e o total dos vencimentos. Por baixo temos duas tabelas, uma com alguns dados do funcionário, como número do BI/CC, NIF, NIB, data de nascimento e morada. A outra tabela mostra todos os vencimentos do funcionário.

Na opção “*Vencimentos*” uma lista em forma de tabela é mostrada com todos os vencimentos realizados até à data. À direita da tabela temos os botões “*informação*” e “*cancelar vencimento*”. No botão “*informação*” são mostradas duas tabelas: uma com os dados do vencimento, salário bruto, bónus, penalização, descontos, estado, observações e salário líquido e a outra com os dados do funcionário.

A opção “*Quartos*” possui um *layout* algo diferente de todos os outros analisados até agora. Do lado esquerdo temos uma coluna com os tipos de quartos que contém três botões. Um botão para “*adicionar novos tipos de quartos*”, outro para “*Adicionar Quarto*” e outro para “*ver dados estatísticos dos quartos*”. Ainda dentro da coluna para cada tipo de quarto existem os botões “*editar*” e “*apagar*”. Do lado direito temos os quartos já criados e o estado actual dos mesmos, se livres ou ocupados.

Ao carregarmos nos botões “*adicionar tipo de quarto*” ou “*Adicionar Quarto*”, um formulário é mostrado ao utilizador. Após o preenchimento dos campos, um novo quarto ou tipo de quarto é adicionado ao sistema. O botão “*estatísticas*” mostra dois gráficos de barras onde se poderá observar o número total de reservas por quarto e a média de noites por quarto.

Ao passarmos à análise dos quartos temos quatro botões; um botão “*fotografias*”, outro com a “*informação*” do quarto, outro para “*editar*” os dados do quarto e, finalmente, o botão “*apagar*” o quarto.

Ao escolhermos a opção “*fotografias*” é mostrado ao utilizador todas as fotografias anexas àquele quarto. Permite, ainda, escolher qual a fotografia principal e quais as que devem estar visíveis ou não. Em baixo, existe um botão para “*adicionar nova fotografia*”.

O botão “*informação*” mostra uma tabela com todas as reservas associadas ao quarto em questão. Em cima temos os dados estatísticos que são: o total de estadias/reservas, total de clientes, total de acompanhantes, média de noites reservadas e facturação total.

Os botões “*editar*” e “*apagar*” o quarto, funcionam à semelhança dos botões com as mesmas funções descritas anteriormente.

Na opção “*Manual*” encontra-se, tal como o nome indica, o manual de utilizador. Em cima, existe uma barra de pesquisa que, ao ser preenchida, vai listando os títulos de um índice que contém as respostas à nossa pesquisa. Por baixo tem um índice que, ao carregar no título, faz aparecer um texto com a descrição de como fazer passo a passo.

A opção “*Front-Office*” encontra-se dividida em três partes: “*Serviços*”, “*Promoções*” e “*Galeria*”.

O utilizador ao escolher “*Serviços*” obtém uma opção com todos os serviços que o empreendimento tem. Em cima encontra-se um botão para adicionar um novo serviço (“*Publicar Serviço*”) e, em baixo, encontramos uma lista, tipo tabela, que mostra uma fotografia do serviço (necessário para o *Front-Office*), o título do serviço, uma descrição e três botões. O primeiro botão serve para adicionar ou editar “*fotografias*”, o segundo botão para “*editar*” o texto do serviço e o terceiro botão para “*apagar*” um serviço. Ao carregar no botão “*editar*” existe a possibilidade do utilizador alterar o estado do serviço para activo ou inactivo.

A opção “*Promoções*” é igual à opção “*Serviços*”. O *layout* é idêntico e tem os mesmos botões de acção.

Ao entrarmos na opção “*Galeria*”, são mostradas todas as fotografias que fazem parte da galeria no *Front-Office*. Para cada imagem existem dois botões, um botão para tornar “*visível ou não fotografia*” e um botão para “*apagar a fotografia*”. Em baixo

surgirá um botão “*adicionar nova fotografia*” e o tipo da fotografia (se esta faz parte das actividades, salas, casa ou quartos).

A última opção do menu lateral é “*Notificações*”. Aqui, uma lista mostrará todas as notificações. Esta lista, em forma de tabela, tem como campos: acção, identificador da acção, data e qual o funcionário que fez a acção.

A opção “*Visitantes*” mostra todos os visitantes que acedem ao *Front-Office*. Em cima temos dados estatísticos como o número total de visitantes, número médio de visitantes por semana, país com mais visitantes e continente com mais visitantes. Em baixo, uma tabela com a data de visita, o IP, o país, a região, a cidade, o continente e a localização (latitude e longitude) é mostrada.

A última opção a ser mostrada é a opção “*Pagamentos*”. Em linha com os outros *layouts*, em cima, temos os dados estatísticos e, em baixo, uma tabela com todos os pagamentos realizados pelos clientes. Os dados estatísticos mostram o total de pagamentos feitos aquando de uma reserva: quanto os clientes pagaram de sinal, quanto pagaram quando fizeram o *check-in*, o gasto em extras por parte dos clientes e o total de facturação (custo da reserva mais extras). Em baixo, a tabela tem como campos: nome do cliente, quando foi feito o pagamento (se na reserva, se aquando do *check-in* ou *check-out*), o valor, total da estadia, tipo de pagamento (monetário, multibanco, etc.) e data do pagamento.

Esta descrição detalhada não engloba todas as funcionalidades da aplicação nem todas as páginas a que os utilizadores têm acesso. Esta descrição é apenas para o leitor ficar com uma ideia geral da homogeneidade dos *layouts* escolhidos. Apesar de a aplicação ter muitas páginas diferentes, percebemos que existe sempre uma preocupação em tentar manter as páginas o mais semelhantes possível. Por norma, os botões com as mesmas funcionalidades são apresentados no mesmo sítio. Verificamos, também, que na maioria das páginas que contém listagens em forma de tabela, por cima destas existem dados estatísticos relativos àquele tema.

5.4. Principais Módulos

5.4.1. *Home*

A página principal e inicial de qualquer aplicação ou *website* tem, por norma, o nome de *homepage*.

Na aplicação, esta *homepage* é a página para a qual os utilizadores, após fazerem o *login*, são conduzidos. Esta página, para além de servir de página de apresentação, contém os dados mais importantes da aplicação e que mais contribuem para o correcto funcionamento da actividade.

Ao introduzir o *email* e *password* na página *web* (*Front-Office*), o sistema verifica se se trata de um utilizador registado, com conta activa e se o *email* e *password* correspondem aos dados introduzidos. Após a confirmação, o sistema verifica de que tipo de utilizador se trata, se utilizador administrador ou utilizador funcionário. Em função do tipo de utilizador, o sistema carrega diferentes *homepages*, *Back-Office* Administração ou *Back-Office* Funcionários.

Ao entrar na aplicação o utilizador é conduzido à sua página principal. Tanto os administradores como os funcionários têm acesso à situação actual de todos os quartos da pousada.

O utilizador pode visualizar todos os quartos da pousada e verificar se estes se encontram ocupados, livres, com *check-out* ou *check-in* pendentes.

Em função dos diferentes estados dos quartos, diferente informação é mostrada ao utilizador. Isto permite uma maior organização e simplicidade da aplicação. Diferentes cores são mostradas para diferentes estados dos quartos; vermelho se o quarto tiver um *check-out* pendente, verde se tiver um *check-in* pendente, azul se estiver ocupado e branco se estiver disponível.

Os estados dos quartos são activados às 00h00 de cada dia em questão. Por exemplo, um quarto com um *check-out* para o dia 3 às 15h00 fica com o botão a vermelho activo às 00h00 deste mesmo dia 3. O mesmo se passa com os outros estados. Os estados de *check-out* e *check-in* encontram-se activos mesmo que as datas não sejam cumpridas. Cabe ao utilizador corrigir estas situações, ou cancelando um *check-in* ou fazendo um *check-out* tardio.

Quando um quarto se encontra ocupado, é mostrado o cliente que o ocupa e, caso tenha acompanhantes, a informação relativa aos mesmos. Os botões informação, extras e *check-out* ficarão activos quando os clientes estão hospedados.

O utilizador ao carregar no botão extras pode adicionar os extras aos pedidos dos clientes. No botão informação é mostrada toda a reserva: o nome do cliente e dos acompanhantes, datas de *check-in* e *check-out*, custo dos extras e outras informações relevantes.

Quando o quarto está ocupado, o botão *check-out* ficará sempre activo para permitir que o cliente possa fazer um *check-out* antecipado, se assim o desejar.

Existe a possibilidade de o mesmo cliente alugar mais do que um quarto mas, como não é possível o cliente estar a dormir simultaneamente em dois quartos, aparece um *icon* a informar qual o quarto em que o cliente não está a dormir, mas do qual fez a reserva.

5.4.2. Reservas

Uma reserva serve para guardar a informação da estadia dos clientes. Para o promotor é muito importante um bom sistema gestor de reservas, pois com ele consegue economizar tempo e aumentar a produtividade. Por este e outros motivos, este módulo é o mais importante e central da aplicação.

Para facilitar, tanto para os clientes como para os funcionários, foi adoptado um método por passos para a inserção de uma reserva. Deste modo, as falhas são minimizadas tanto a nível dos utilizadores como a nível do sistema. A cada passo, o sistema avalia se o passo anterior está preenchido correctamente. Antes de escrever na base de dados o sistema volta a verificar a disponibilidade do quarto e só depois guardará os dados.

Ao introduzir uma reserva, esta encontra-se no estado de pré-reserva, o primeiro de quatro estados. Uma reserva pode assumir cinco estados (Figura 6).

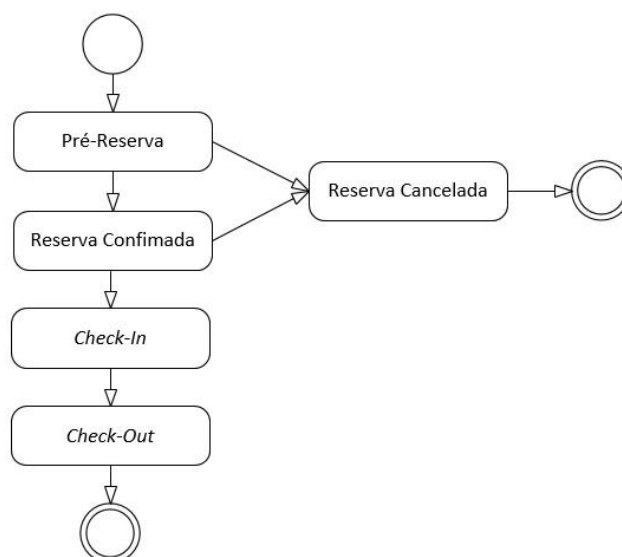


Figura 6 - Diagrama de estados das Reservas

O estado de pré-reserva garante o bloqueio da disponibilidade do quarto. Deste modo, não será possível existir uma nova reserva para aquele quarto nas datas em que já se encontra reservado. (Anexo I - Tabela 4 – Requisito Qrt-08)

O funcionário, ao confirmar uma reserva, altera o estado da mesma, deixando de ser uma pré-reserva e passando a uma reserva confirmada (Anexo I – Tabela 11 - Requisito Res-10).

Para confirmar uma reserva é necessário que o cliente já tenha depositado um sinal na conta da empresa. Este valor era, inicialmente, de 50%, mas acabou por ser alterado para um valor variável. Esta alteração deve-se ao facto de diferentes casas de turismo terem diferentes taxas de reservas, permitindo ainda aplicar taxas diferentes a clientes diferentes.

Ao confirmar uma reserva, o quarto aparece no calendário da aplicação com as datas de *check-in* e *check-out* (Anexo I – Tabela 12 - Requisito Cal-02).

Quando o cliente chega à pousada, o estado da reserva é alterado para *check-in* ou hospedado após o cliente realizar o pagamento do remanescente da reserva.

Ao ser alterada a estadia para *check-in*, o sistema modifica o estado do quarto para ocupado e activa a funcionalidade “*Extras*”. Esta funcionalidade permitirá ao hóspede consumir produtos da pousada ou da casa de turismo, sendo o valor creditado na sua conta.

Antes do cliente abandonar a pousada ou casa de turismo, o funcionário terá de alterar o estado da reserva para *check-out*. Ao fazer isto, o sistema informará se existe algum extra pendente e qual o valor do mesmo.

O sistema, ao receber a confirmação de *check-out*, automaticamente altera o estado do quarto para livre, ficando este disponível para nova reserva.

No caso de cancelamento, só será possível cancelar uma reserva nos seus dois primeiros estados (pré-reserva ou reserva confirmada). O cancelamento de uma reserva é, por norma, um acontecimento ocasional e, por isso, apenas é possível nestes dois estados. Também não é normal um cancelamento no estado de *check-in*/hospedado, pois no caso de o cliente sair antecipadamente, é efectuado um *check-out* antes do tempo.

Todas as reservas e respectivos estados (pré-reserva, reserva confirmada, reserva finalizada, hospedado e reserva cancelada) podem ser consultadas no “*Histórico*”.

Tanto os clientes como os funcionários podem realizar uma nova reserva. Os clientes fazem uma nova reserva na página principal (*Front-Office*), enquanto que os funcionários fazem uma nova reserva na aplicação (*Back-Office*).

No *Front-Office* foi dada mais ênfase à facilidade, à rapidez e ao aspecto de todos os passos necessários para confirmar uma pré-reserva.

No *Front-Office* era de grande importância a facilidade e rapidez com que o utilizador fazia uma nova reserva.

Todo este módulo foi desenvolvido com esse intuito, tendo sido escolhido um sistema por passos em que o utilizador vai avançando à medida das suas necessidades.

À semelhança do *Back-Office*, se o utilizador já se encontra registado, apenas tem de preencher o campo do *email*, em vez de fazer um novo registo.

No *Back-Office* foi enfatizada a rapidez e a qualidade da informação, em detrimento do aspecto. Para um funcionário introduzir uma reserva é necessário que o cliente já exista na base de dados.

Antes de ser criada uma nova reserva na base de dados, o sistema analisa todos os dados que recebeu e verifica se os quartos pretendidos ainda estão disponíveis. Este passo é necessário para garantir que, caso dois clientes estejam a fazer uma reserva ao mesmo tempo, a reserva que irá ser criada será a do cliente que submeteu primeiro os dados.

5.4.3. Hóspedes

Clientes

Um dos principais objectivos desta aplicação foi recolher o máximo possível de informação relativa ao cliente. Deste modo será possível, mais tarde, tratar esses dados e fazer, por exemplo, campanhas de *marketing*, promoções, entre outros.

Para tratar os dados pessoais dos clientes, estes são guardados em duas fases:

- Registo.
- *Check-In*.

Aquando do registo, o sistema guarda o dado principal que consiste no *email*. Este dado funciona como o identificador único do cliente, pois não existem dois clientes como o mesmo *email*, apesar de ser possível efectuar um registo sem *email* do cliente. Outros dados relevantes são guardados no registo, como o nome, idade, morada, nacionalidade, contacto, entre outros. (Anexo I – Tabela 13 - Requisito Cli-03).

Como foi descrito no módulo anterior, um cliente pode ser registado de duas formas: pelo próprio cliente ou por um funcionário. O registo efectuado pelo cliente é feito no

módulo *Front-Office* e obrigatoriamente tem de fornecer o *email*. Um registo feito por um funcionário, não necessita de fornecer este campo. (Anexo I – Tabela 13 - Requisito Cli-02).

Quando o cliente faz o *check-in* na pousada, são solicitados novos dados complementares: o número do bilhete de identidade e o número de identificação fiscal. O sistema verifica se já existem clientes com o mesmo número de bilhete de identidade e identificação fiscal e, se for o caso, não permite acrescentar estes novos dados ao cliente. (Anexo I – Tabela 13 - Requisito Cli-03).

Acompanhantes

Os acompanhantes ou convidados são todos os hóspedes que não são clientes. Isto é, são os hóspedes que vêm junto com os clientes mas cuja reserva não está em seu nome.

Os acompanhantes são registados aquando do *check-in* e, à semelhança dos clientes, o *email* é usado como identificador único. Se o convidado não tiver *email* é criado um convidado novo. (Anexo I – Tabela 11 - Requisito Res-22).

Como só é possível existir acompanhantes caso exista um cliente, coloca-se o problema do acompanhante voltar à pousada com um cliente diferente e ocorrer a duplicação de dados. Para que isto não aconteça, o sistema verifica se o acompanhante já existe na base de dados. Se for este o caso, o sistema apenas faz um *update* aos dados existentes, caso contrário, cria um acompanhante novo.

Cada cliente ou acompanhante tem uma ficha de reservas (Figura 7) e, aqui, podem ser visualizadas todas as reservas dos clientes, assim como os gastos por estadia e a média de noites que estiveram na pousada.

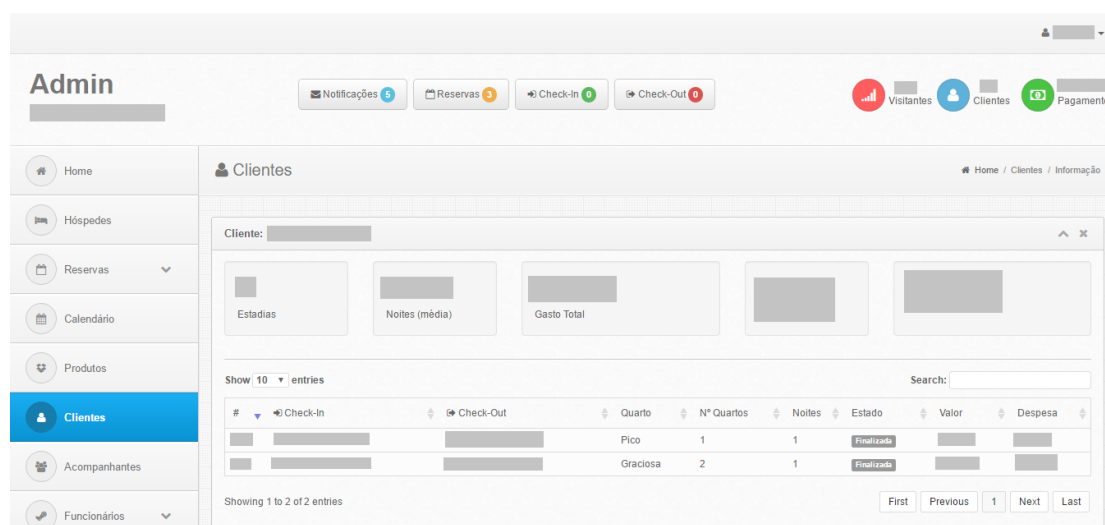


Figura 7 - Aplicação / Informação Clientes

5.4.4. Quartos

Dada a importância que os quartos têm em todo o negócio, este é um dos módulos com mais funcionalidades implementadas.

Na aplicação desenvolvida era importante existir a distinção das épocas alta, normal e baixa. Para além do valor dos quartos ser diferente, deveria ainda ser possível definir as datas dessas épocas. Quando é criada uma reserva, o sistema verifica a respectiva data e, consoante, qual o valor que deve atribuir, consoante seja época alta, baixa ou normal.

Outra verificação importante do sistema consiste na avaliação da capacidade dos quartos. Foi criado um algoritmo para verificar a capacidade de um quarto para hospedar um certo número de adultos e/ou crianças. Para o correcto funcionamento do algoritmo, foi criada uma tabela temporária que vai guardando o número de adultos designados para um quarto e o mesmo para o número de crianças.

Se a um quarto lhe for designado um número de adultos ou crianças superior ao permitido, o sistema não permite avançar o processo de criação de uma reserva. (Figura 8).

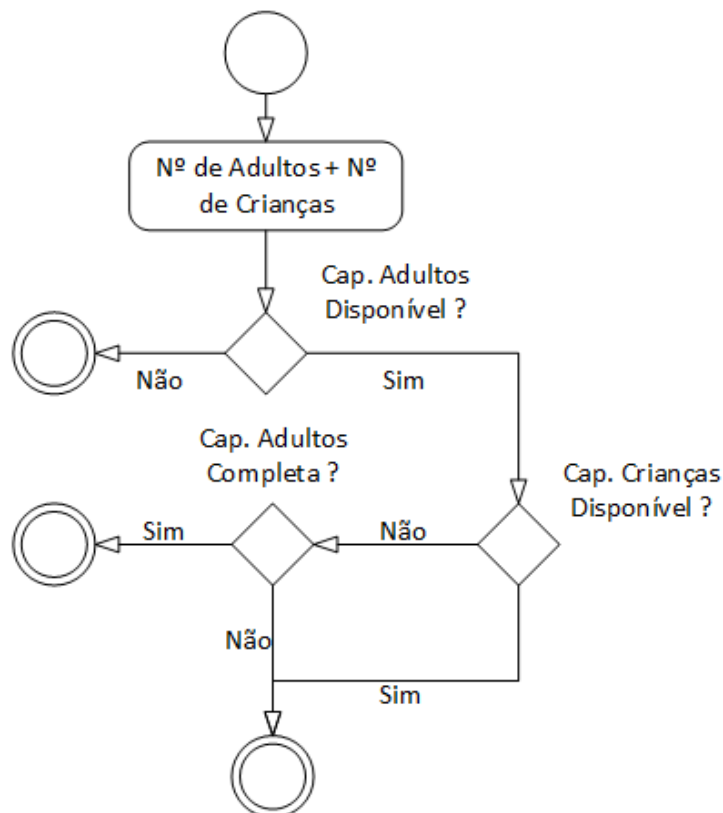


Figura 8 - Diagrama de actividade - Algoritmo verifica capacidade

Outra verificação consiste em testar a capacidade de um quarto para adulto com criança. Uma criança pode ficar num quarto com disponibilidade para adultos, mas um adulto não pode ficar num quarto com disponibilidade para uma criança.

Ao serem mostrados dados estatísticos (como o número de reservas, o número de clientes e acompanhantes, o número de noites e a facturação total) é possível realizar uma melhor análise dos quartos e, consequentemente, a tomada de decisões mais acertadas.

Adicionalmente, também é possível visualizar através de dois gráficos de barras o total de reservas e o número total de noites por quarto.

Para além de permitir adicionar, alterar e remover um quarto, este módulo funciona directamente com o *Front-Office*. Como tal, a introdução dos dados tem de ser mais cuidada, pois estes irão aparecer na página *web*.

Ao introduzir um novo quarto, o sistema verifica se já existe algum quarto com o mesmo nome. Se for este o caso, um alerta é mostrado ao utilizador e o novo quarto não é criado. (Anexo I - Tabela 4 - Requisito Qrt-01).

Inicialmente estava previsto a página *web* ter apenas uma *view* para os quartos com uma descrição detalhada e várias fotos. No entanto, foi criada uma outra funcionalidade, a de adicionar fotografias por quarto. Assim, a cada quarto poderão ser associadas uma ou mais fotos. Ao serem adicionadas várias fotografias, será necessário definir qual a fotografia (principal) que irá surgir no segundo passo da reserva (*Front-Office*), e se as fotografias introduzidas estão activas ou não para aparecer, aquando da reserva.

Ao introduzir uma fotografia, esta encontra-se inactiva por omissão, fazendo com que não apareça directamente no *Front-Office*.

Dentro deste módulo também existe a possibilidade de introduzir o tipo de quarto existente. Um cliente, ao fazer uma reserva, terá de indicar que tipo de quarto está a reservar (e.g. quarto *single*, *twin*, *double*, *triple*, ou outro). Para isso, foi criado todo um conjunto de funcionalidades para adicionar, remover e editar o tipo de quarto.

Ao criar um novo tipo de quarto, este irá aparecer no módulo *Front-Office* acompanhado de uma pequena descrição.

5.4.5. Funcionários

Neste módulo, podemos observar as características de todos os funcionários e os respectivos pagamentos.

Para além de ser possível listar todos os funcionários que trabalham na pousada, também é possível adicionar novos funcionários, fazendo a distinção entre administrador ou funcionário normal. (Anexo I – Tabela 2 - Requisito Fun-02).

O *email* do funcionário é o identificador único para o sistema, fazendo com que não seja possível existirem dois funcionários com o mesmo *email*, ou com o *email* de um cliente já registado.

Outra funcionalidade implementada são os dados estatísticos dos funcionários. Aqui podemos observar o número de acessos à aplicação, o número de reservas, *check-ins*, *check-outs* e produtos vendidos, fazendo com que o dono da aplicação tenha uma percepção da actividade individual de cada funcionário.

A cada funcionário também se encontra associada uma lista de todos os pagamentos mensais. Aqui, o administrador poderá lançar novos pagamentos aos seus funcionários, fazendo esta informação aparecer na página individual de cada um. (Anexo I – Tabela 3 - Requisito Ven-02).

Sempre que é lançado um novo pagamento, o sistema fica encarregue de efectuar os cálculos sobre os descontos, deixando apenas para o utilizador a introdução do salário bruto e da percentagem de descontos a atribuir.

Foi adoptada a funcionalidade de não deixar apagar um vencimento lançado. Apenas é possível cancelar o lançamento. Foi adoptado este método para garantir que não existam discrepâncias ou ocultações de valores.

5.4.6. Utilizadores

Os utilizadores são todos aqueles que, de alguma maneira, interagem com a aplicação.

O sistema tem três tipos de utilizadores: os clientes, através do *Front-Office*, os administradores e os funcionários, através do *Back-Office*.

Os utilizadores do *Back-Office* possuem um papel muito importante na aplicação, uma vez que serão os utilizadores principais. Este módulo pode ser dividido em duas categorias:

- O Utilizador Administrador.
- O Utilizador Funcionário.

Inicialmente apenas um tipo de utilizador, o Utilizador Administrador, era contemplado. Visto a aplicação ser destinada a uma pequena casa de turismo rural, o utilizador seria sempre o dono da mesma e, por isso, não faria sentido a distinção entre administrador e funcionário.

No entanto, a aplicação faz esta distinção, encontrando-se dividida em duas categorias.

Ao entrar na aplicação, o sistema verifica se se trata de um funcionário ou administrador, direccionando os diferentes tipos de utilizadores para as suas respectivas páginas principais.

O Utilizador Administrador é um super utilizador tendo acesso a toda a aplicação. Ele gere todos os módulos instalados e só ele pode adicionar novos administradores ou funcionários.

Apenas o Utilizador Administrador pode fazer a gestão dos módulos “*Quartos*”, “*Vencimentos*”, “*Funcionários*”, “*Pagamentos*” e “*Notificações*”.

Este super utilizador tem de ser criado directamente na base de dados aquando da instalação da aplicação.

A necessidade da escrita directamente na base de dados do super utilizador deve-se ao facto de, para se entrar na aplicação, ser necessário um actor creditado para tal.

Só depois da primeira entrada na aplicação, é possível criar novos funcionários, novos quartos, reservas, etc.

O Utilizador Funcionário é um utilizador com restrições. Este utilizador não tem acesso a vários módulos, ficando limitado aos módulos essenciais para gerir as reservas.

Os módulos a que os funcionários têm acesso são: “*Home*”, “*Hóspedes*”, “*Reservas*”, “*Calendário*”, “*Produtos*”, “*Clientes*”, “*Acompanhantes*”, “*Manual de Utilizador*” e “*Front-Office*”. Este utilizador apenas pode ser criado pelo administrador do sistema.

O Utilizador Cliente interage com a aplicação no *Front-Office* ficando limitado apenas à possibilidade de adicionar uma nova reserva.

5.4.7. **Front-Office**

Por definição, o *Front-Office* consiste na camada da aplicação que se encontra mais próxima do cliente sendo, de uma maneira geral, a página *web* principal de uma empresa.

Normalmente estas páginas *web* têm muitos conteúdos estáticos, sendo necessário, caso se pretenda mudar apenas um título, entrar directamente no ficheiro e alterar manualmente o que se pretende.

Como o *Front-Office* iria ter vários conteúdos dinâmicos não seria exequível estar frequentemente a entrar directamente nos ficheiros e alterar o que se pretendia. Assim, foi desenvolvida uma componente específica na aplicação para lidar com estas alterações.

Existem quatro sub-menus para lidar com o *Front-Office*:

- Galeria.
- Quartos.
- Serviços / Actividades.
- Promoções.

A galeria é um espaço destinado exclusivamente à adição e remoção das fotografias que irão surgir na “*Galeria*” da página *web* (*Front-Office*). Aqui são mostradas as fotografias da casa, dos quartos, das salas e das actividades.

Todas as fotografias adicionadas terão de ter uma altura de 300px e largura de 200px, para não desformatar a página *web* e poderem ser agrupadas em grupos de três por linha.

Para além de ser possível fazer o *upload* das imagens para o servidor, também é possível apagá-las.

Ao fazer o *upload*, o utilizador seleccionará a fotografia e irá anexá-la ao grupo a que pertence (e.g. casa, salas, actividades ou quartos). O nome das imagens será o identificador das mesmas para o sistema, não permitindo a existência de duas fotografias com o mesmo nome. (Anexo I - Tabela 5 - Requisito Fro-18).

Os formatos permitidos são: GIF, JPG, PNG e JPEG.

Outra funcionalidade das imagens é a possibilidade de estarem visíveis ou não. O utilizador pode fazer o *upload* das imagens para o servidor, e escolher se estas serão mostradas ou não no *Front-Office*.

5.4.8. Produtos

Outra fonte de receitas num hotel são os extras consumidos pelos hóspedes.

Estes são, normalmente, produtos de consumo ou até mesmo prestações de serviços.

A criação do módulo “*Produtos*” foi realizada com o intuito do utilizador da aplicação ter uma visão geral dos consumos internos dos seus hóspedes, e não tanto pela vertente financeira do custo de venda de cada produto.

Outro objectivo consistiu na facilidade de adicionar um extra a uma estadia.

O utilizador ao adicionar um novo produto terá de colocar uma referência única, sendo esta o identificador único do produto.

Se o utilizador se enganar ao introduzir os dados, o sistema detecta automaticamente que aquele produto já existe e não permite o avanço da acção. (Anexo I - Tabela 1 - Requisito Pro-03).

Existe a possibilidade de desactivar um produto sempre que este fica descontinuado ou quando o administrador/funcionário não quiser que apareça na lista de venda. Para tal, existe um campo que permite desactivar um produto. (Anexo I - Tabela 1 - Requisito Pro-07).

Um produto é vendido no módulo “*Home*”.

Sempre que um quarto se encontra ocupado, um *icon* referente aos produtos fica activo. Ao carregar neste *icon* surgem todos os produtos activos, de entre os quais o utilizador pode escolher o desejado pelo cliente.

Para adicionar um produto a uma reserva basta escolher a quantidade. O sistema irá automaticamente adicionar este produto e o seu custo à reserva.

Outro campo importante do produto é a quantidade existente em armazém. Não é possível vender uma quantidade de produtos superiores à existente em armazém. (Anexo I - Tabela 1 - Requisito Pro-09).

Para dados estatísticos, são mostradas individualmente todas as reservas nas quais aquele produto foi consumido, assim como as quantidades vendidas e a receita total.

5.4.9. Pagamentos

É normal os hotéis ou pequenas pousadas solicitarem um adiantamento ou sinal das suas reservas. Assim, garantem que o quarto estará ocupado para as datas pretendidas e que a desistência por parte do cliente seja menos provável.

Todas as reservas estão associadas aos pagamentos e, na aplicação desenvolvida, estes encontram-se divididos em três etapas:

- Sinal (confirmação de reserva).
- *Check-In*.
- Extras (*Check-Out*).

Quando é feita uma reserva, esta encontra-se no estado de pré-reserva e só passa ao estado de reserva quando é feito o primeiro pagamento (sinal).

Após o cliente efectuar o pagamento e o utilizador efectivar a confirmação do mesmo, o sistema guarda o valor do sinal para que, no momento do *check-in*, seja deduzido este valor ao valor da reserva.

Valor da reserva = valor da diária do quarto x número de noites.

Valor do sinal = percentagem do valor da reserva.

Valor em falta = valor da reserva – valor do sinal.

Quando o cliente chega à pousada, ao fazer o *check-in* terá de pagar o remanescente do valor em falta. O sistema determina este valor automaticamente e o utilizador apenas tem de confirmar que recebeu o valor.

No final do *check-in* o valor da reserva está pago na totalidade, isto é, o cliente antes de entrar para o quarto tem a sua estadia paga para os dias que contratualizou.

Ao fazer o *check-out* o cliente pode, ou não, ter de pagar o valor dos extras. Sempre que o cliente consome um extra (produtos ou serviços) o valor é acrescentado à reserva do cliente. Estes extras são pagos quando o cliente faz o *check-out*, cabendo ao sistema alertar o utilizador para a existência de extras pendentes.

Só depois do utilizador introduzir o valor dos extras é que o sistema avança e finaliza toda a reserva. O cliente poderá, então, abandonar o hotel ou pousada com todos os pagamentos saldados.

Na aplicação é mostrada uma lista com todos os pagamentos feitos, o nome do cliente que os fez, as etapas a que correspondem e, entre outros dados, o valor discriminado. (Anexo I – Tabela 8 - Requisito Pag-01).

Para dados estatísticos é mostrado o valor total dos pagamentos, o valor total dos pagamentos iniciais (sinal), o valor total aquando do *check-in* e o valor total dos pagamentos relativos aos extras. (Anexo I – Tabela 8 - Requisito Pag-02).

5.4.10. Calendário

O calendário é uma ferramenta fundamental para qualquer sistema gestor de hotelaria. Ele facilita a identificação dos quartos que estão disponíveis para uma determinada data e também mostra as datas em que os quartos estão ou estiveram ocupados.

Visto tratar-se de um componente com alguma importância, foi adoptada a API Full Calendar.

Por se tratar de uma API, existiram algumas restrições que limitaram as potencialidades inicialmente previstas para o calendário. A maior restrição foi o facto de não ser permitido criar uma reserva directamente no calendário.

Em contrapartida, a visualização das reservas dos quartos é muito simples e intuitiva.

Adicionalmente, a navegação pelos meses é muito fácil e a possibilidade de atribuir uma cor a cada quarto facilita ainda mais a identificação rápida da disponibilidade destes.

Para este módulo foi criada uma tabela SQL única, a tabela “*Calendário*”. Esta tabela é responsável por guardar todos os dados necessários para o correcto funcionamento da API.

Sempre que é feita uma reserva, o sistema insere os novos dados na tabela calendário. Estes dados são inseridos quando um cliente ou funcionário cria uma pré-reserva, fazendo com que o quarto apareça automaticamente no calendário. (Anexo I – Tabela 12 - Requisito Cal-02).

Quando é feito o cancelamento da pré-reserva ou da reserva, o quarto é automaticamente retirado do calendário, ficando disponível para nova reserva.

5.4.11. Visitantes

Outro requisito importante consiste em saber quais os mercados que mais consultavam os seus serviços. Para isso foi criado o módulo “*Visitantes*”.

Este módulo detecta quem entra na página *web* principal e, através do seu IP, descodifica a informação do país de origem e da cidade.

Adicionalmente, também é possível verificar quantos visitantes acedem à página principal (*Front-Office*) da aplicação. Sempre que um novo IP consulta a página principal é incrementado o total de visitantes. Foi criado um algoritmo que através do IP verifica se um dado utilizador já consultou a página principal naquele dia e, se for o caso, não incrementa o valor “*Visitantes*”.

Através de outro algoritmo também é possível obter a informação da média de visitantes por semana, o país e o continente que mais consulta a página *web*.

5.4.12. Notificações

Como o *Back-Office* irá ter mais do que um utilizador, será importante a documentação do maior número de acções que os utilizadores fazem.

Este módulo apenas se encontra visível para os Utilizadores Administradores, permitindo visualizar o tipo de acções que os seus funcionários mais executam e ter documentadas as datas de inserções de valores.

Das muitas acções possíveis foram escolhidas 14:

- Entrada na aplicação.
- Criado novo cliente.
- Feita pré-reserva.
- Confirmação de reserva.
- Efectuado *Check-In*.
- Efectuado *Check-Out*.
- Reserva cancelada.
- Venda de produtos.
- Cliente editado.
- Criado novo produto.
- Produto editado.
- Acompanhante editado.

Sempre que o Utilizador Administrador entra na aplicação aparece um alerta com as novas notificações, encontrando-se estas sempre visíveis durante toda a navegação pela aplicação.

Adicionalmente a serem mostradas as acções dos utilizadores, também são mostrados os números identificadores dos atributos que as acções alteram. Por exemplo, se a acção é cancelar uma reserva, é mostrado o número identificador da reserva que foi cancelada. Assim, é mais simples para o administrador rastrear os atributos que são afectados pelas acções dos utilizadores.

6. Testes e Resultados

Para qualquer *software*, um dos factores mais importantes, se não o mais importante, é a sua permeabilidade a erros.

Sendo esta uma aplicação construída de raiz, os erros detectados foram alguns e nem sempre as metodologias de implementação de código foram as melhores.

Foi necessário chegar-se à fase de testes para serem verificadas metodologias de implementação erradas. Por exemplo, inicialmente uma reserva era adicionada à medida que o utilizador avançava nos passos. Se, no último passo, o utilizador em vez de finalizar uma reserva clicasse noutro *link*, a reserva ficaria registada com dados incorrectos ou incompletos.

Outros erros foram detectados devido a falhas na análise de requisitos. Um dos exemplos é o caso de um hóspede não ter *email* para fornecer. Se assim fosse o hóspede não poderia ser registado no sistema, tornando impossível a criação de um histórico referente a este mesmo cliente.

Após a implementação de cada módulo eram feitos testes sem planeamento ou documentação (*Ad hoc*). Estes testes consistiam em, através de um fluxo normal, determinar se o utilizador obtinha ou não os resultados pretendidos. Se sim, o teste seguinte era através de um fluxo alternativo, normalmente uma inserção errada, verificando se o sistema respondia correctamente ao erro introduzido. Sempre que o teste retornava um resultado que não fosse o esperado, o módulo era novamente reprogramado até ter a correcta implementação.

Foram inseridos registos para simular aproximadamente um ano de actividade da empresa. Esta simulação serviu, principalmente, para testar os *layouts* e se estes, ao ficarem carregados de informação, não desviavam o utilizador das tarefas que pretendia executar.

Através de um teste de usabilidade informal, foi solicitado a cinco utilizadores, que não conheciam a aplicação, que a usassem a durante aproximadamente vinte minutos para verificar se conseguiam inserir uma reserva, fazer o *check-in* e *check-out* de um cliente e adicionar extras à reserva.

O utilizador de testes tinha de fazer as seguintes tarefas.

- Inserir um cliente.
- Fazer uma pré-reserva.

- Confirmar uma reserva.
- Fazer o *Check-In*.
- Adicionar um extra à reserva.
- Fazer o *Check-Out* de reservas já existentes.

Todos os utilizadores conseguiram fazer todas as tarefas e nenhum necessitou dos vinte minutos para as concluir.

Este resultado demonstrou que a aplicação desenvolvida vai ao encontro de um dos principais objectivos propostos, que era a facilidade de utilização por parte de novos utilizadores.

7. Conclusões

Os objectivos traçados foram cumpridos tendo os mesmos, em alguns casos, ido além do inicialmente previsto.

Apesar de ter sido feita uma análise de requisitos cuidada, durante o desenvolvimento da aplicação detectou-se que existiam alguns módulos importantes que não estavam contemplados no trabalho inicial.

Durante o desenvolvimento do projecto, certas funcionalidades demoraram muito mais tempo a ser implementadas do que previsto. Funcionalidades que, aparentemente, seriam de baixa dificuldade revelaram ser de difícil implementação. Outras vezes era exactamente o oposto, funcionalidades que supostamente seriam difíceis, mostraram ser de simples implementação.

Com o avançar dos trabalhos e do tempo, ocorreu uma evolução tanto como programador como empreendedor do projecto de turismo. Apesar da escolha de uma metodologia em cascata pouco aberta a mudanças, foi relativamente simples fazer com o que o projecto se fosse moldando à referida evolução, sem necessitar de grandes mudanças estruturais.

Este projecto poderá ser considerado uma versão 1.0 de um sistema gestor de hotéis de pequena escala. Não se encontra fechado e novos módulos serão implementados à medida que for necessário. Esta foi também uma das principais razões pelo qual este projecto foi escolhido, dada a possibilidade de adicionar, alterar ou remover módulos à medida das necessidades.

Num futuro próximo, novas funcionalidades serão implementadas, uma das quais será a possibilidade de o registo dos clientes ser realizado através de serviços externos, como o Facebook.

Antes do projecto Quinta dos Mistérios - Turismo de Habitação ser iniciado comercialmente, irá ser integrado também um sistema (já desenvolvido no trabalho final de curso) de *check-lists* que será usado para verificar o estado dos quartos e se estes cumprem os padrões exigidos para albergar novos clientes.

Ao desenvolver a aplicação, pretendeu-se que esta não fosse muito focada no negócio Quinta dos Mistérios – Turismo de Habitação, precisamente com o intuito da sua comercialização. Optou-se por desenvolver uma aplicação genérica, de modo a ser usada por outras casas de turismo de habitação ou *hostels*.

Para poder comercializar esta aplicação, a escolha da metodologia de instalação por módulos revelou ser a mais eficaz. Desta forma, o cliente final pode escolher os módulos que deseja comprar e adaptá-los ao seu modelo de negócio.

O resultado final revelou algumas diferenças deste produto em relação aos existentes no mercado. Este sistema gestor contém os módulos que o cliente entende serem os mais importantes para o desenvolvimento da sua actividade.

Outra diferença encontrada é que são poucas as aplicações existentes que fazem referência aos acompanhantes dos hóspedes e poucas guardam os seus dados. Adicionalmente a funcionalidade de reservas para grupos (+ de 1 quarto) não é muito comum nos aplicativos existentes e testados.

Outro factor é a importância que é dada ao cliente.

Ao utilizar a página *web* desta aplicação, é transmitida a sensação ao cliente de que faz parte do negócio, não sendo apenas mais um número.

Podemos apontar como pontos fortes:

- *Login* possível em qualquer local que disponha de acesso à Internet.
- Existência de um *web-site* conectado ao *Back-Office*.
- Criação e gestão dos clientes.
- Criação e gestão dos acompanhantes.
- Acesso a vários dados estatísticos.
- Possibilidade de fazer reservas para grupos.
- Sistema de visualização de *Check-In* e *Check-Out* para os próximos dias.
- Possibilidade de fazer reservas *online* por parte dos clientes.
- Sistema de aviso de novas reservas por *email*.
- *Layout* intuitivo e de fácil utilização.
- Criação e gestão dos quartos.
- *Mobile Access*.
- Possibilidade de adicionar conteúdos ao *Front-Office*, como fotografias, serviços e promoções.
- Possibilidade de definir épocas altas, baixas ou normais.

Relativamente aos pontos fracos, referem-se:

- A falta de integração em canais de distribuição, como agências de viagens *online*.
- Não existência da possibilidade de promoções (tipo *voucher* ou regalias).
- Não existência de um módulo limpeza dos quartos.
- Calendário pouco desenvolvido e sem possibilidade de fazer reservas directamente.
- Impossibilidade de imprimir dados estatísticos.
- Inexistência de um sistema interno de mensagens.
- Não estar disponível noutras línguas.
- A falta de experiência do promotor pode revelar que nem todos os módulos necessários possam estar implementados.

Para além de uma interrupção, de alguns meses, para frequentar um curso de empreendedorismo, houve períodos em que não foi possível cumprir o planeamento. Uma das maiores dificuldades encontradas foi a falta de uma equipa com a qual discutir ideias e possíveis implementações de funcionalidades.

8. Bibliografia

1. *Revista Hotéis*. [1 de Maio de 2007] edição 55. Softwares são imprescindíveis para gestão hoteleira. Disponível na Internet em:
< <http://www.revistahoteis.com.br/softwares-sao-imprescindiveis-para-gestao-hoteleira/>
2. *SINERGY*. [Março de 2010] newsletter nº3. Tecnologias da Informação na Indústria Hoteleira: Visão de um Gestor Hoteleiro. Disponível na Internet em:
<<http://www.sinfic.com/SinficWeb/displayconteudo.do2?numero=47612&artigo=47603>>
3. The Top 5 Free and Open Source Hotel Management Software Solutions.
[Publicado a 18 de Fevereiro 2016]. Disponível na Internet em:
< <http://blog.capterra.com/top-free-and-open-source-hotel-management-software-solutions-compared/> >
4. The Top 5 Most Popular Hotel Management Software for Small Hotels Compared.
[Publicado a 30 de Agosto 2016]. Disponível na Internet em:
< <http://blog.capterra.com/the-5-most-popular-hotel-management-software-solutions-for-small-hotels-compared/> >
5. Abacre Cloud Hotel Management System.2014. Disponível na Internet em:
< <http://www.abacre.com/cloudhotelmanagementsystem/> >
6. HotelRunner Hotel Management System. 2016. Disponível na Internet em:
< <http://hotelrunner.com/en/> >
7. Base7Booking Hotel Management System. 2016. Disponível na Internet em:
< <https://www.base7booking.com/> >
8. eZee FrontDesk Hotel Management Software. 2016. Disponível na Internet em:
< <http://www.ezeefrontdesk.com/> >
9. RoomMaster Standard Edition Hotel Management Software. 2016. Disponível na Internet em:
< <http://roommasterpms.co.uk/> >
10. HotelOgix Smart Hoteliering. 2016. Disponível na Internet em:
< <http://www.hotelogix.com/portuguese/> >
11. Hoteliga Clowd Hotel Management System. 2016. Disponível na Internet em:
< <http://www.hoteliga.com/> >
12. Framework Bootstrap. 2016. Mark Otto and Jacob. Disponível na Internet em:
< <http://getbootstrap.com/2.3.2/> >

13. Framework Codeigniter.[Última actualização Março 2016]. Disponível na Internet em:
< http://www.codeigniter.com/user_guide/ >
14. ZingChart API. 2016. Disponível na Internet em:
< <https://www.zingchart.com/> >
15. Full Calendar API. 2016. Disponível na Internet em:
< <http://fullcalendar.io/> >
16. Compare Hotel & Hospitality Management Software. 2016. Disponível na Internet em:
< <http://www.softwareadvice.com/hotel-management/?more=true#more> >
17. PHP documentation. 2016. The PHP group. Disponível na Internet em:
< <https://secure.php.net/docs.php> >
18. Stack Overflow. Community of 4.7 million programmers. 2016 Disponível na Internet em:
< <http://stackoverflow.com/> >
19. TAVARES, Frederico. 11/2012. Desenvolvimento de Aplicações em PHP.
20. Top Hospitality Property Management Software Products.2016. Disponível na Internet em:
< <http://www.capterra.com/hospitality-property-management-software/>
21. Universidade Lusíada do Porto (Guia para elaboração de dissertações de mestrado) 2007. Disponível na Internet em:
<http://www.por.ulusiada.pt/downloads/ilpg/REGULAMENTO_ELABORACAO_MESTRADOS_DOUTORAMENTOS.pdf >
22. Wrapbootstrap.Bootstrap themes and templates.2016. Disponível na Internet em:
< <https://wrapbootstrap.com/> >
23. W3schools.com. PHP 5 tutorial.2016. Disponível na Internet em:
< <http://www.w3schools.com/php/default.asp> >

9. Anexos

9.1. Anexo I - Lista de Requisitos

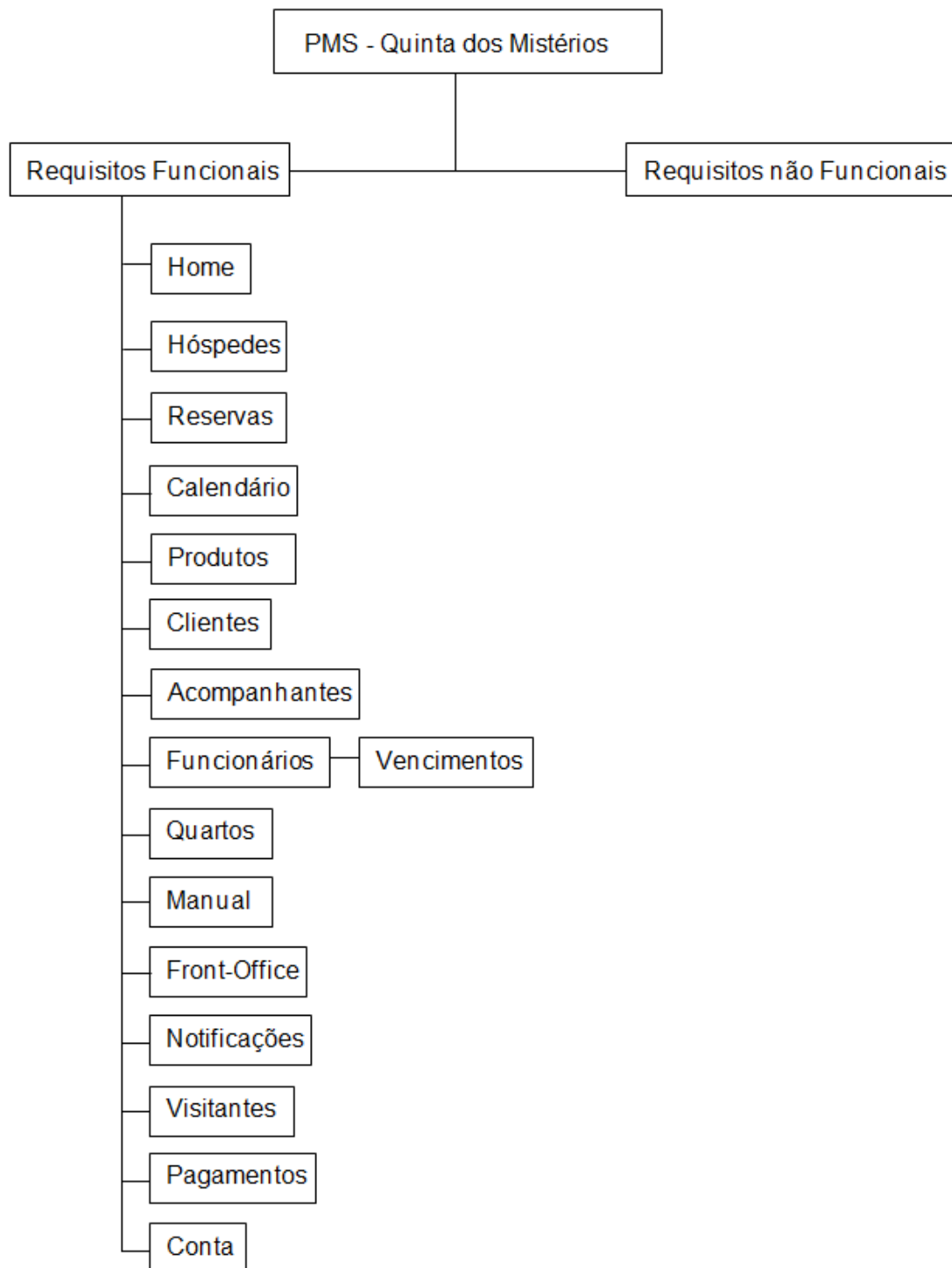
Este documento tem duas funções: ser um documento informativo para a equipa do projecto (relativamente a todos os requisitos necessários à implementação da aplicação) e ser um documento informativo para o dono da aplicação (que fica a saber se esta vai ao encontro das suas necessidades).

Estes requisitos foram definidos através de contactos com clientes e com docentes da Escola Superior de Tecnologia e Gestão de Oliveira do Hospital.

As tabelas de requisitos estão divididas por módulos para facilitar o entendimento dos mesmos. Cada tabela tem como colunas: **código do requisito**, **descrição**, **prioridade**, **risco** e **implementação**.

- Código do requisito (Cód. Req.) - Identificador único do requisito.
- Descrição - Relato textual do requisito.
- Prioridade (Prio.) - O grau de prioridade serve como escala para a implementação de um requisito em que 1 é o grau mais elevado (requisito com grande importância de implementação) e 5 o menos elevado (requisito com muito pouca importância de implementação).
- Risco - O risco servirá para alertar o impacto que terá a má implementação ou a não implementação de um requisito. 1 é o grau mais elevado (má implementação pode conduzir a uma falha muito grave) e 5 o grau menos elevado (má implementação pode conduzir a uma pequena falha).
- Implementação (Impl.) – Esta coluna informa se o requisito foi ou não codificado (implementado) na aplicação.

9.1.1. Visão geral dos requisitos



9.1.2. Requisitos Funcionais.

9.1.2.1. Módulos Utilizador Administrador.

9.1.2.1.1. Módulo Produtos.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Pro-01	Deve ser possível visualizar uma lista com todos os produtos. Os campos que devem aparecer são: nome, marca, referência, quantidade, local no armazém, descrição, estado e preço.	1	1	Sim
Pro-02	O sistema tem de permitir o registo de novos produtos. Os campos são: modelo/nome, marca, referência, tipo, descrição, preço, quantidade e local no armazém.	1	1	Sim
Pro-03	O sistema deve garantir que a referência é única. Não podem existir dois produtos com a mesma referência. Todos os campos são de preenchimento obrigatório, com excepção da descrição.	1	1	Sim
Pro-04	Quando um produto é adicionado com sucesso no sistema, um alerta deve ser mostrado ao administrador.	3	3	Sim
Pro-05	O administrador pode editar os dados de um produto. Ao alterar a referência o sistema deve verificar se essa referência já existe e, se assim for, não permitir a edição deste campo.	1	1	Sim
Pro-06	Deve ser possível ver a informação dos produtos e alguns dados estatísticos. Deve ser possível visualizar todas as estadias em que o produto foi vendido, quantidade total de vendas e valor, assim como outras informações relevantes, para além dos dados do produto.	2	1	Sim

Pro-07	O administrador pode desactivar um produto. Quando um produto é desactivado (bloqueado), deixa de poder ser vendido.	1	1	Sim
Pro-08	Um produto só pode ser apagado da aplicação se não tiver dados associados a ele. Se existirem dados associados ao produto, este apenas pode ser desactivado (bloqueado).	1	1	Sim
Pro-09	Um produto só pode ser vendido se tiver <i>stock</i> positivo.	1	1	Sim
Pro-10	Sempre que um produto é vendido a quantidade é decrementada no <i>stock</i> e o valor é acrescentado à reserva do cliente.	1	1	Sim

Tabela 1 - Módulo Produtos

9.1.2.1.2. Módulo Funcionários

Cód. Req.	Descrição	Prio.	Risco	Impl.
Fun-01	Deve ser possível visualizar uma lista com todos os funcionários, Os campos que devem aparecer são: tipo (administrador ou funcionário), nome, <i>email</i> , contacto, data de registo, número de acessos e estado.	1	1	Sim
Fun-02	O sistema tem de permitir o registo de novos funcionários. Os campos são: nome, sobrenome, data de nascimento, <i>email</i> , contacto, morada, BI/CC, NIF, NIB e tipo (administrador ou funcionário). Todos os campos são de preenchimento obrigatório.	1	1	Sim
Fun-03	Aquando do registo de um funcionário por parte da administração, o sistema tem de garantir que o <i>email</i> é único.	1	1	Sim

Fun-04	O sistema deve verificar se o NIF introduzido é válido.	1	1	Não
Fun-05	Quanto um funcionário é introduzido com sucesso no sistema, um alerta deve ser mostrado ao administrador.	1	1	Sim
Fun-06	O administrador pode editar os dados do funcionário. Se alterar o <i>email</i> , o sistema tem de verificar se esse <i>email</i> ainda não existe para validar a alteração.	1	1	Sim
Fun-07	Só o funcionário na sua página pessoal pode alterar a <i>password</i> , todos os outros dados podem ser alterados por parte do administrador.	1	1	Sim
Fun-08	Deve ser possível ver a informação dos funcionários e alguns dados estatísticos. Deve ser possível visualizar o total de acessos à aplicação, data/hora dos acessos, número de <i>check-ins</i> e <i>check-outs</i> , quantidade de produtos vendidos, total de vencimentos e outras informações relevantes, para além dos dados pessoais.	1	1	Sim
Fun-09	O administrador pode desactivar um funcionário. Quando um funcionário é desactivado (bloqueado), deixa de poder entrar na aplicação.	1	1	Sim
Fun-10	Um funcionário só pode ser apagado da aplicação se não tiver dados associados a ele. Se existirem dados associados ao funcionário este apenas pode ser desactivado (bloqueado).	1	1	Sim

Tabela 2 - Módulo Funcionários

9.1.2.1.3. Módulo Vencimento Funcionários.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Ven-01	Deve ser possível visualizar uma lista com todos os pagamentos efectuados aos funcionários e administradores. Os campos que devem aparecer são: data do pagamento, nome do funcionário ou administrador, salário bruto, bónus, penalização, descontos, estado e salário líquido.	1	1	Sim
Ven-02	O sistema tem de permitir o registo de novos pagamentos. Os campos são: nome, salário bruto, bónus, penalizações, descontos e observações. Todos campos são de preenchimento obrigatório, à excepção das observações.	1	1	Sim
Ven-03	Quando um pagamento é introduzido com sucesso no sistema um alerta deve ser mostrado ao administrador.	3	3	Sim
Ven-04	O administrador pode editar os dados de um pagamento.	1	1	Sim
Ven-05	Deve ser possível ver a informação de um pagamento a um funcionário. Todos os dados dos pagamentos devem ser mostrados, assim como os dados do funcionário.	1	1	Sim
Ven-06	O sistema deve calcular automaticamente o salário líquido do funcionário.	2	1	Sim
Ven-07	Um vencimento não pode ser apagado do sistema. Apenas é possível cancelar um vencimento. Um vencimento cancelado não aparece nos pagamentos dos funcionários (módulo <i>Conta</i>).	2	1	Sim

Tabela 3 - Módulo Vencimento Funcionários

9.1.2.1.4. Módulo Quartos.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Qrt-01	O sistema deve permitir adicionar novos quartos. Só o administrador o pode fazer. Os campos são: nome, valor, atributos (se tem vistas, tv, <i>wi-fi</i> e <i>wc</i>), tipo (<i>single/double/twin, etc</i>), descrição, capacidade de adultos, capacidade de crianças, cor de fundo, preço época alta e época baixa, início da época alta e da baixa e fim da época alta e da baixa. Não deve ser possível adicionar um novo quarto com o mesmo nome de um quarto já existente.	1	1	Sim
Qrt-02	Deve existir uma verificação para não permitir campos em branco e para que as datas do fim das épocas (alta e baixa) não possam ser anteriores às datas de início.	1	1	Sim
Qrt-03	Quando um quarto é adicionado com sucesso, um alerta deve ser mostrado ao utilizador.	3	3	Sim
Qrt-04	Deve ser possível editar todos os campos dos quartos. O sistema deve garantir que as datas do fim das épocas (alta e baixa) não possam ser anteriores às datas de início.	1	1	Sim
Qrt-05	O administrador pode desactivar um quarto. Quando um quarto é desactivado deixa de aparecer no módulo <i>Home</i> e não pode ser reservado.	1	1	Sim
Qrt-06	Deve ser possível visualizar todos os quartos e as suas especificações. Ao mostrar os quartos deve existir uma indicação se estão ocupados ou livres.	1	1	Sim

Qrt-07	Um quarto só pode ser removido se não tiver dados associados a ele. Caso tenha, só pode ser desactivado.	1	1	Sim
Qrt-08	Se um quarto está reservado, não pode permitir novas reservas para aquele período de tempo.	1	1	Sim
Qrt-09	Se um quarto estiver ocupado, não pode permitir reservas para aquele período de tempo.	1	1	Sim
Qrt-10	Um quarto deixa de estar reservado quando a reserva é cancelada ou quando o cliente faz <i>check-out</i> .	1	1	Sim
Qrt-11	Um quarto passa ao estado de livre quando um cliente faz <i>check-out</i> .	1	1	Sim
Qrt-12	Quanto um cliente faz <i>check-in</i> o quarto passa a estar ocupado.	1	1	Sim
Qrt-13	Deve ser possível ver a informação dos quartos e alguns dados estatísticos. Deve ser possível visualizar todas as estadias em que o quarto esteve reservado, número total de estadias, número total de clientes e acompanhantes, média de noites e a facturação total.	2	2	Sim
Qrt-14	O sistema deve permitir adicionar novos tipos de quartos. Só o administrador pode adicionar novos tipos de quartos. Os campos são: tipo de quarto (<i>single/double/twin,etc</i>) e descrição.	1	1	Sim

Qrt-15	Não deve ser possível adicionar um tipo de quarto que já exista.	1	2	Sim
Qrt-16	Deve ser possível editar o tipo de quarto. O sistema deve garantir que não seja possível alterar para um tipo que já exista.	1	1	Sim
Qrt-17	Uma janela modal deve ser mostrada quando um novo tipo de quarto é criado. Também deve ser mostrada uma janela modal se o tipo de quarto for alterado.	3	3	Sim
Qrt-18	Só deve ser possível apagar um tipo de quarto se não existir nenhum quarto associado a esse tipo. Uma janela modal deve ser mostrada a informar o utilizador desta opção.	1	1	Sim
Qrt-19	Antes de apagar um tipo de quarto, uma janela modal deve ser mostrada ao utilizador confirmando se deseja mesmo apagar este tipo de quarto.	1	1	Sim
Qrt-20	Deve ser possível adicionar imagens aos quartos. Cada quarto pode ter várias imagens.	2	1	Sim
Qrt-21	As imagens dos quartos devem ter o tamanho máximo de 500Kb, altura de 280px e largura de 185px.	3	4	Sim
Qrt-22	Deve ser possível activar e desactivar as imagens que se querem mostrar.	1	1	Sim
Qrt-23	Deve existir a possibilidade de o utilizador escolher qual a imagem principal a ser mostrada no módulo <i>Front-Office</i> .	3	1	Sim

Qrt-24	Uma janela modal deve ser mostrada ao utilizador, antes de este apagar uma imagem, perguntando se deseja mesmo apagá-la.	4	1	Sim
Qrt-25	Se uma imagem estiver visível não é possível apagar esta imagem.	1	1	Sim
Qrt-26	O utilizador ao escolher uma imagem para principal, esta, se estiver inactiva, passa automaticamente para activa.	2	2	Sim

Tabela 4 - Módulo Quartos

9.1.2.1.5. Módulo *Front-Office*.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Fro-01	Deve ser possível adicionar novos serviços à página <i>web</i> principal. Os campos devem ser: título a vermelho, título, descrição e imagem.	1	1	Sim
Fro-02	Ao adicionar um novo serviço, o sistema tem de garantir que não existem duas imagens com o mesmo nome.	1	1	Sim
Fro-03	O mesmo serviço não pode ter mais do que uma imagem.	1	1	Sim
Fro-04	Uma lista com todos os serviços deve ser mostrada. Os campos a mostrar são: imagem, título a vermelho, título, descrição, data de inserção e estado.	1	1	Sim
Fro-05	Deve ser possível editar um serviço e todos os campos podem ser alterados.	1	1	Sim

Fro-06	Um serviço quando criado fica no estado activo. Deve ser possível alterar o estado de activo para inactivo. Quando um serviço estiver inactivo não aparece na página <i>web</i> inicial.	1	1	Sim
Fro-07	Antes de eliminar um serviço, uma janela modal deve ser mostrada ao utilizador a perguntar se este deseja mesmo apagar o serviço. Quando um serviço é eliminado, este deixa de poder aparecer na página <i>web</i> da aplicação.	3	3	Sim
Fro-08	Deve ser possível adicionar novas promoções à página <i>web</i> principal. Os campos devem ser: título a vermelho, título, descrição e imagem.	1	1	Sim
Fro-09	Ao adicionar uma nova promoção, o sistema tem de garantir que não existem duas imagens com o mesmo nome.	1	1	Sim
Fro-10	A mesma promoção não pode ter mais do que uma imagem.	1	1	Sim
Fro-11	Uma lista com todas as promoções deve ser mostrada. Os campos a mostrar são: imagem, título a vermelho, título, descrição, data de inserção e estado.	1	1	Sim
Fro-12	Deve ser possível editar uma promoção, todos os campos podem ser alterados.	1	1	Sim
Fro-13	Uma promoção, quando criada, fica no estado activa. Deve ser possível alterar o estado de activa para inactiva. Quando uma promoção estiver desactivada não aparece na página <i>web</i> inicial.	1	1	Sim

Fro-14	Antes de eliminar uma promoção, uma janela modal deve ser mostrada ao utilizador a perguntar se este deseja mesmo apagar o serviço. Quando uma promoção é eliminada esta deixa de poder aparecer na página <i>web</i> da aplicação.	3	3	Sim
Fro-15	As imagens dos serviços e das promoções devem ter o tamanho máximo de 500Kb, altura de 200px e largura de 120px.	2	1	Sim
Fro-16	Deve ser possível adicionar fotografias ou imagens para divulgar a empresa (<i>Galeria</i>).	2	1	Sim
Fro-17	As imagens para divulgar a empresa são divididas em 4 tipos: Casa, Quartos, Salas e Actividades. As imagens devem ter o tamanho máximo de 500Kb, altura de 300px e largura de 200px	2	1	Sim
Fro-18	O sistema tem garantir que não existem duas imagens com o mesmo nome na secção <i>Galeria</i> .	1	1	Sim
Fro-19	Deve ser possível eliminar uma imagem. Se uma imagem for eliminada, esta deixa de aparecer na página <i>web</i> da aplicação.	1	1	Sim
Fro-20	Deve ser possível alterar o estado de uma imagem de visível para não visível. Uma imagem não visível deixa de aparecer na página <i>web</i> da aplicação.	1	1	Sim

Tabela 5 - Módulo *Front-Office*

9.1.2.1.6. Módulo Notificações.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Not-01	Quando um cliente deixa o seu <i>feedback</i> , uma nova notificação é criada.	2	1	Não
Not-02	Quando um artigo/produto está com o <i>stock</i> próximo do fim, uma notificação deve ser mostrada.	4	1	Não
Not-03	<p>Todas as notificações são mostradas na área da aplicação reservada ao administrador. Uma lista com todas as notificações deve ser mostrada. Os campos a mostrar são: acção, ID da acção, data e nome do funcionário. O campo ID acção é relativo ao identificador único do atributo do módulo que é alterado. Por exemplo:</p> <p>Acção - confirmar reserva, ID da acção – reserva número 335.</p> <p>Acção - entrada na aplicação, ID da acção – funcionário número 4.</p>	1	1	Sim
Not-04	O sistema deve guardar as seguintes notificações: entrada na aplicação por parte de um utilizador, criação de novo cliente, nova pré-reserva, confirmação de uma reserva, novo <i>check-in</i> , <i>check-out</i> , cancelamento de uma reserva, criação de novo funcionário, criação de novo administrador, novo produto vendido, produto editado e acompanhante/convidado editado.	1	1	Sim

Tabela 6 - Módulo Notificações

9.1.2.1.7. Módulo Visitantes.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Vis-01	Deve existir uma lista com todos os utilizadores que acederam à página <i>web</i> da aplicação. Os campos a serem mostrados devem ser: data, IP, país, região, cidade, continente, latitude e longitude.	1	1	Sim
Vis-02	Deve existir um contador com o número total de visitantes da página <i>web</i> . Este contador tem de garantir que o mesmo visitante não é incrementado se visitar a página mais do que uma vez no mesmo dia.	1	1	Sim
Vis-03	Devem existir funcionalidades para mostrar a média de visitantes por semana e qual o país e o continente que mais visita a página.	1	1	Sim

Tabela 7 - Módulo Visitantes

9.1.2.1.8. Módulo Pagamentos.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Pag-01	Deve existir uma lista com todos os pagamentos efectuados pelos clientes. Os campos mostrados devem ser: nome do cliente, altura em que o pagamento é feito, valor, total da estadia, tipo de pagamento e data.	1	1	Sim
Pag-02	Devem existir funcionalidades para mostrar o total de pagamentos feitos na reserva (sinal), no <i>check-in</i> e no <i>check-out</i> (extras).	1	1	Sim

Tabela 8 - Módulo Pagamentos

9.1.2.2. Módulos Utilizador Administrador/Funcionário.

9.1.2.2.1. Módulo Home.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Hom-01	Na página inicial devem aparecer todos os quartos activos da aplicação (ocupados ou livres).	1	1	Sim
Hom-02	Se o quarto se encontra ocupado deve aparecer uma indicação a azul, se estiver livre deve aparecer a branco.	1	2	Sim
Hom-03	Os quartos devem ter a possibilidade de, no imediato, mostrarem quais os clientes que os ocupam e respectiva data de entrada e saída.	1	1	Sim
Hom-04	Quando os quartos estão ocupados pelos acompanhantes/convidados e não pelo cliente deve existir um alerta a mostrar qual o cliente responsável pela reserva.	3	1	Sim
Hom-05	Se existirem acompanhantes, um <i>link</i> deve ficar activo para mostrar a informação referente aos mesmos.	2	2	Não
Hom-06	Devem existir <i>links</i> para mostrar a informação da reserva, compras de produtos e para fazer <i>check-out</i> .	1	1	Sim
Hom-07	Se a estadia tiver produtos associados (Extras), um alerta deve ser mostrado.	1	1	Sim
Hom-08	Se o prazo de <i>check-out</i> tiver expirado, um alerta deve ser mostrado.	2	2	Sim

Home-09	Deve ser possível visualizar o estado dos quartos nos dias seguintes.	2	2	Sim
Home-10	Se o quarto se encontrar vazio os <i>links</i> para as informações, <i>check-out</i> e extras devem estar inactivos.	1	1	Sim
Home-11	Se o quarto se encontra vazio um botão para criar uma reserva deve existir.	1	1	Sim
Home-12	Se o quarto tiver um <i>check-out</i> para aquele dia, um botão com um atalho para o <i>check-out</i> deve existir.	1	1	Sim
Home-13	Se o quarto tiver um <i>check-in</i> para aquele dia, um botão com um atalho para o <i>check-in</i> deve existir.	4	4	Não

Tabela 9 - Módulo *Home*

9.1.2.2.2. Módulo Hóspedes.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Hos-01	Deve ser possível ver uma lista com os clientes que estão hospedados no hotel. Esta lista deve mostrar qual o tipo (cliente ou acompanhante), o nome, o quarto, o <i>email</i> , o contacto e a data de <i>check-out</i>).	1	1	Sim
Hos-02	Os acompanhantes/convidados dos clientes também devem ser mostrados nesta lista.	1	2	Sim
Hos-03	Se possível, os acompanhantes devem estar agrupados com os respectivos clientes.	3	3	Sim

Tabela 10 - Módulo Hóspedes

9.1.2.2.3. Módulo Reservas.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Res-01	O método de adicionar uma pré-reserva deve ser feito por passos.	3	3	Sim
Res-02	<p>Passo 1 (escolha do cliente): Deve aparecer uma lista com os clientes registados e activos na aplicação. O sistema deve garantir que apenas um cliente possa ser seleccionado.</p>	1	1	Sim
Res-03	<p>Passo 2 (dados da estadia): Os campos a preencher são: data de <i>check-in</i>, data de <i>check-out</i>, número de adultos, número de crianças, <i>transfer</i> e pedidos especiais. O sistema deve garantir que o campo data de <i>check-in</i> não seja anterior à data actual e o campo data de <i>check-out</i> não seja anterior à data de entrada. Todos os campos, à excepção do campo pedidos especiais, são obrigatórios.</p>	1	1	Sim
Res-04	<p>Passo 3 (escolha dos quartos): Ao utilizador deve aparecer uma lista com os quartos disponíveis para a data pretendida. O sistema tem de garantir que os quartos escolhidos estão livres para as datas pretendidas assim como se têm a capacidade para albergar o número de adultos e crianças. O sistema deve ainda garantir que não sejam escolhidos mais quartos do que os necessários. (2 adultos não poderão reservar 3 quartos, por exemplo).</p>	1	1	Sim

Res-05	<p>Passo 4 (resumo): Antes de adicionar a pré-reserva, devem ser mostrados ao utilizador (tipo folha de pagamento) os dados do cliente, as datas de <i>check-in</i> e <i>check-out</i>, o quarto, capacidade e tipo de quarto, número de noites, preço por noite, época (alta normal ou baixa)) e o total da estadia. Depois da pré-reserva adicionada com sucesso, uma janela modal deve aparecer a informar o utilizador.</p>	1	1	Sim
Res-06	<p>Deve existir uma lista com todas as reservas <u>não confirmadas</u> (pré-reservas). Devem ser mostrados os campos: <i>check-in</i>, <i>check-out</i>, número de hóspedes, número de quartos, data da criação da pré-reserva e cliente.</p>	1	1	Sim
Res-07	<p>O utilizador deve poder consultar a informação de uma reserva <u>não confirmada</u>. A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05).</p>	1	1	Sim
Res-08	<p>Deve ser possível cancelar uma reserva <u>não confirmada</u>. Ao cancelar uma reserva o seu estado é alterado e pode ser acedida em reservas canceladas, deixando de aparecer na lista das reservas não confirmadas.</p>	1	1	Sim
Res-09	<p>Deve existir uma lista com todas as reservas <u>confirmadas</u>. Devem ser mostrados os campos: <i>check-in</i>, <i>check-out</i>, número de hóspedes, número de quartos, data da criação da pré-reserva e cliente.</p>	1	1	Sim

Res-10	Antes de confirmar a reserva um formulário com valor monetário (sinal) deve ser preenchido. Este formulário tem como campos: valor do sinal e método de pagamento (monetário, multibanco, cheque, visa, ou transferência bancária). A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05) e é obrigatório inserir o valor do sinal.	1	1	Sim
Res-11	Após confirmar a reserva, um <i>email</i> deve ser enviado ao cliente para o informar da mesma.	3	1	Sim
Res-12	O utilizador deve poder consultar a informação de uma reserva <u>confirmada</u> . A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05).	1	1	Sim
Res-13	Deve ser possível cancelar uma reserva <u>confirmada</u> . Ao cancelar uma reserva o seu estado é alterado e pode ser acedida em reservas canceladas deixando de aparecer na lista das reservas confirmadas.	1	1	Sim
Res-14	Deve existir uma lista com todas as reservas <u>canceladas</u> . Devem ser mostrados os campos: <i>check-in</i> , <i>check-out</i> , estado, número de quartos, data do cancelamento, número de ocupantes e cliente.	4	4	Não
Res-15	O utilizador deve poder consultar a informação de uma reserva <u>cancelada</u> . A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05).	1	1	Sim

	Deve existir uma lista com o histórico das reservas. Devem ser mostrados os campos:			
Res-16	<i>check-in</i> , <i>check-out</i> , estado, número de quartos, data da acção, número de hóspedes e cliente.	1	1	Sim
Res-17	Deve existir uma lista com todas as reservas <u>confirmadas que dão entrada no hotel no dia</u> (<i>check-in</i>). Devem ser mostrados os campos: quarto, cliente, <i>check-in</i> , <i>check-out</i> , número de hóspedes e número de quartos.	1	1	Sim
Res-18	Só deve ser possível adicionar uma reserva se existirem quartos disponíveis para as datas pretendidas.	1	1	Sim
Res-19	Antes de escrever os dados na base de dados o sistema deve verificar se o quarto ainda se encontra disponível, assim não há o risco de existirem duas reservas para o mesmo quarto em simultâneo.	1	1	Sim
Res-20	O método de <i>check-in</i> deve ser feito por passos.	3	3	Sim
Res-21	Passo 1 (cliente): Deve aparecer uma lista com os dados do cliente e os campos BI/CC e NIF. Estes campos devem ser preenchidos, caso ainda não tenham sido. O sistema deve garantir que o número do BI/CC e do NIF não existem noutros clientes.	1	1	Sim
Res-22	Passo 2 (Acompanhantes): Se existirem acompanhantes, dados como o nome, o <i>email</i> , o tipo (criança ou adulto), o contacto e a data de nascimento devem ser preenchidos. Se o <i>email</i> do acompanhante já existe, não é adicionado na base de dados, se não existe é adicionado. Se não existirem acompanhantes o utilizador “salta” para o passo 3.	1	1	Sim

Res-23	<p>Passo 3 (Distribuição dos quartos): Os clientes e acompanhantes devem ser distribuídos pelos quartos reservados. O sistema deve garantir que todos os quartos reservados são seleccionados. O sistema deve verificar se a capacidade dos quartos corresponde aos hóspedes (ex: um quarto com capacidade para um adulto e uma criança não pode acomodar dois adultos. Um quarto com capacidade para dois adultos pode acomodar um adulto e uma criança).</p>	1	1	Sim
Res-24	<p>Passo 4 (Pagamento): Antes de finalizar o <i>check-in</i> um formulário com valor monetário (pagamento) aparece já preenchido. O utilizador apenas seleccionará o método de pagamento (monetário, multibanco, cheque, visa ou transferência). A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05) e com o campo valor em falta (total da reserva – valor do sinal). O sistema automaticamente introduz este valor em falta.</p>	1	1	Sim
Res-25	<p>Uma janela modal deve aparecer a informar o utilizador que o <i>check-in</i> foi efectuado com sucesso.</p>	3	2	Sim
Res-26	<p>Deve existir uma lista com todas as reservas <u>confirmadas que dão saída do hotel no dia (<i>check-out</i>)</u>. Devem ser mostrados os campos: quarto, cliente, <i>check-out</i>, número de hóspedes, número de quartos e extras. Os dados só aparecem no dia do <i>check-out</i>.</p>	1	1	Sim

Res-27	Antes de confirmar o <i>check-out</i> um formulário com valor monetário (pagamento) deve ser preenchido se o cliente tiver feito alguma despesa. Este formulário tem como campos: valor do pagamento e método de pagamento (monetário, multibanco, cheque, visa ou transferência). A reserva deve ser mostrada ao utilizador (tipo folha de pagamento, à semelhança do requisito Res-05) e com o campo valor em falta (<u>extra</u>). O sistema deve confirmar se o valor introduzido perfaz o valor do <u>extra</u> .	1	1	Sim
Res-28	O sistema deve apresentar os valores da reserva consoante as épocas em que esta é feita (preços de época baixa, normal e época alta).	1	1	Sim
Res-29	Sempre que apareça um quarto, o utilizador deve visualizar a que época os valores se referem (alta, baixa ou normal).	1	1	Sim
Res-30	Uma lista com todas as reservas deve ser mostrada (<i>Histórico</i>) para o utilizador poder ver toda a informação de uma reserva.	1	1	Sim
Res-31	A informação de uma reserva deve ser mostrada. Deve ser possível ver as datas em que o estado da reserva foi alterado (pré-reserva, reserva confirmada, <i>check-in</i> e <i>check-out</i>), os clientes e acompanhantes dessa reserva, datas de <i>check-in</i> , <i>check-out</i> , número de noites, <i>transfer</i> , pedidos especiais, número de hóspedes, qual o quarto ou quartos reservados, quais os produtos consumidos e o total da despesa.	1	1	Sim

Tabela 11 - Módulo Reservas

9.1.2.2.4. Módulo Calendário.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Cal-01	Os quartos reservados devem ser mostrados numa página tipo calendário.	1	1	Sim
Cal-02	Sempre que uma reserva é confirmada, as datas de início e fim devem mostradas no calendário.	1	1	Sim

Tabela 12 - Módulo Calendário

9.1.2.2.5. Módulo Clientes.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Cli-01	Deve ser possível visualizar uma lista com todos os clientes. Os campos que devem aparecer são: nome, <i>email</i> , contacto, registo e estado.	1	1	Sim
Cli-02	O sistema tem de permitir o registo de novos clientes. Os campos são: nome, sobrenome, data-nascimento, <i>email</i> , contacto, morada, cidade, distrito, código-postal e país.	1	1	Sim
Cli-03	O sistema deve garantir que o <i>email</i> , o BI e o NIF são únicos. Não podem existir dois clientes com o mesmo <i>email</i> . Todos os campos são de preenchimento obrigatório, à excepção do <i>email</i> . Se o cliente não possuir <i>email</i> , deve ser adicionado em branco.	1	1	Sim
Cli-04	Quando um cliente é adicionado com sucesso no sistema, um alerta deve ser mostrado ao utilizador.	2	2	Sim
Cli-05	O utilizador pode editar os dados de um cliente. Ao alterar o <i>email</i> o sistema deve verificar se já não existe um cliente com esse mesmo <i>email</i> .	1	1	Sim

	Deve ser possível ver a informação dos clientes e alguns dados estatísticos. Deve ser possível visualizar todas as estadias que o cliente reservou, número médio de noites, valor total dos seus gastos, assim como outras informações relevantes, para além dos dados pessoais.	1	1	Sim
Cli-06				
	O administrador pode desactivar um cliente. Quando um cliente é desactivado (bloqueado), deixa de poder entrar na aplicação e deixa de aparecer na lista das reservas.	1	1	Sim
Cli-07				
	Um cliente só pode ser apagado da aplicação se não tiver dados associados a ele. Se existirem dados associados ao cliente, este apenas pode ser desactivado (bloqueado).	1	1	Sim
Cli-08				
	Se um cliente estiver bloqueado e fizer uma reserva através do <i>Front-Office</i> um alerta deve ser mostrado ao utilizador de <i>Back-Office</i> .	1	1	Não
Cli-09				

Tabela 13 - Módulo Clientes

9.1.2.2.6. Módulo Acompanhantes/Convidados

Cód. Req.	Descrição	Prio.	Risco	Impl.
Aco-01	Uma lista com todos os acompanhantes deve ser mostrada. Os campos são: nome, <i>email</i> , contacto e data de registo.	1	1	Sim
Aco-02	Deve ser possível visualizar toda a informação de um acompanhante como, total de estadias, média de dormidas, quarto, cliente e outras informações relevantes.	1	1	Sim

Aco-03	O convidado só pode ser criado no <i>check-in</i> . O sistema só permite criar o mesmo número de convidados contratualizados aquando da pré-reserva.	1	1	Sim
Aco-04	Os campos a preencher ao criar um novo convidado são: nome, <i>email</i> , contacto, tipo (adulto ou criança) e data de nascimento. Os campos nome e data de nascimento são de preenchimento obrigatório.	1	1	Sim
Aco-05	O sistema tem de garantir que não existem dois convidados com o mesmo <i>email</i> .	1	1	Sim
Aco-06	Um convidado que não contenha <i>email</i> pode ser adicionado à base de dados.	1	1	Sim
Aco-07	Não é possível eliminar um convidado.	3	1	Sim
Aco-08	Ao editar um convidado, o sistema tem de garantir que aquele <i>email</i> não existe na base de dados.	1	1	Sim

Tabela 14 - Módulo Acompanhantes/Convidados

9.1.2.2.7. Módulo Manual.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Man-01	Deve existir um manual de utilizador dentro da aplicação, com a descrição de como fazer as principais acções.	1	1	Sim
Man-02	Uma barra de procura deve existir para facilitar as pesquisas do utilizador.	1	1	Sim

Tabela 15 - Módulo Manual

9.1.2.2.8. Módulo Conta.

Cód. Req.	Descrição	Prio.	Risco	Impl.
Cot-01	Deve ser possível, a todos os utilizadores, a alteração de alguns dos seus dados pessoais. Os dados que podem ser alterados são: contacto, morada e <i>password</i> .	1	1	Sim
Cot-02	Deve ser possível, a todos os utilizadores, a visualização dos seus dados pessoais como: nome, <i>email</i> , contacto, morada, BI/CC, NIF, NIB, data de nascimento, data de registo, tipo (administrador ou funcionário) e estado.	1	1	Sim
Cot-03	Deve ser possível, a todos os utilizadores, a visualização dos seus pagamentos. Os campos a serem mostrados são: data, salário bruto, bónus, penalização, descontos, estado, observações e salário líquido.	2	1	Sim

Tabela 16 - Módulo Conta

9.1.3. Requisitos Não Funcionais

Cód. Req.	Descrição	Prio.	Risco	Impl.
Nfunc-01	A aplicação irá ser desenvolvida em PHP, <i>JQuery</i> , CSS, HTML e SQL.	3	3	Sim
Nfunc-02	Os dados serão tratados através de uma base de dados relacional.	3	3	Sim
Nfunc-03	O sistema de reservas deve funcionar por passos. Pré-Reserva, Reserva Confirmada, <i>Check-In</i> , <i>Check-Out</i> .	2	2	Sim
Nfunc-04	Os valores numéricos devem ter entre 0 e 2 casas decimais.	2	2	Sim

Tabela 17 - Requisitos Não Funcionais

9.2. Anexo II - Base de Dados

Neste anexo são explicadas, resumidamente, as funcionalidades das tabelas da base de dados e as principais relações entre elas.

Entidades:

Acompanhantes: Entidade responsável por guardar os dados pessoais dos hóspedes que acompanham o cliente. O atributo *tipo de acompanhante* é para distinguir se se trata de um adulto ou criança.

{AC_ID: int; AC_NOME: varchar; AC_TIPO: varchar; AC_EMAIL: varchar; AC_CONTACTO: varchar; AC_DATA_NASC: date; AC_DATA_REG: date & time}

Quartos: Entidade que contém todos os dados dos quartos. Esta entidade é uma das principais da aplicação e está relacionada com outras 8 entidades. Serve para, entre outras coisas, associar os clientes, os acompanhantes e as reservas a determinado quarto.

{QR_ID: int; QR_NOME: varchar; QR_VALOR: decimal; QR_VISTA: smallint; QR_TV: smallint; QR_WIFI: smallint; QR_DISPONIBILIDADE: smallint; QR_DATA_CRIACAO: date & time; QR_PRECO_ALTA: decimal; QR_PRECO_BAIXA: decimal; QR_DATA_ALTA_INI: date; QR_DATA_ALTA_FIM: date; QR_DATA_BAIXA_INI: date; QR_WC: smallint; QR_NUM_OCUP: smallint; QR_NUM_OCUP_C: smallint; QR_COR: varchar; QR_ESTADO: smallint}

Quartos_Acomp: Entidade que guarda os dados não pessoais dos acompanhantes e faz a ligação com as reservas, quartos e clientes. Esta entidade foi criada para saber qual o quarto atribuído ao acompanhante. Como para existir um acompanhante tem de existir um cliente e uma reserva, foi feita a ligação desta tabela às tabelas estadias (reservas) e clientes. De notar que, um acompanhante pode ficar em quartos diferentes e pode vir a acompanhar clientes diferentes, podendo também verificar-se o inverso. Por isso foram criadas as relações de N para N.

{QA_ID: int; AC_ID: int; QR_ID: int; ES_ID: int; QA_DATA: date & time; CL_ID: int; AC_REL: string}

Clientes: É uma das entidades principais da aplicação, pois guarda os dados pessoais dos clientes. Esta entidade tem uma grande importância, pois sem ela não era possível o funcionamento de outras entidades relevantes, como as estadias. Para existir uma estadia é necessário existir um cliente. Todas as entidades relacionadas com a estadia são dependentes da existência de um cliente.

```
{CL_ID: int; CL_NOME: varchar; CL_SOBRENOME: varchar; CL_EMAIL: varchar;
CL_TELEFONE: varchar; CL_MORADA: varchar; CL_CIDADE: varchar; CL_REGIAO:
varchar; CL_COD_POST: varchar; CL_PAIS: varchar; CL_PASSWORD: varchar;
CL_DATA_REG: date & time; CL_ACESSOS: int; CL_DATA_NASC: date; CL_CC: varchar;
CL_NIF: varchar}
```

Quartos_Cli: Entidade semelhante à entidade Quartos_Acomp. Esta tabela foi criada para saber em que quarto o cliente pernoita. Como existe a possibilidade do cliente alugar mais do que um quarto, é necessário saber em que quarto o cliente fica.

```
{QC_ID: int; CL_ID: int; ES_ID: int; QR_ID: int; QC_DATA: varchar}
```

Dados_Estadia: Esta entidade é necessária para saber quantos quartos o cliente irá alugar. Quando o utilizador faz uma reserva e o aluguer dos quartos é superior a um, é necessário guardar essa informação. Também é necessário guardar o valor das épocas alta, normal e baixa de cada quarto, para que não existam discrepâncias nos dados.

```
{ES_ID: int; CL_ID: int; QR_ID: int; DE_VALOR: decimal; DE_EPOCA: varchar;
DE_VALOR_EPOCA: decimal; DE_CAP: varchar; DE_TIPO: varchar}
```

Feedback: Inicialmente estava projectada a possibilidade de os clientes, no final da estadia, poderem dar o seu *feedback*. Apesar desta funcionalidade não ter sido implementada, uma tabela foi criada com as relações necessárias à implementação desta funcionalidade.

```
{FB_ID: int; ES_ID: int; CL_ID: int; FB_DATA: date & time; FB_ESTADO: smallint; FB_TEXT:
varchar}
```

Estadia: Tabela central da aplicação. Esta entidade guarda as reservas feitas pelos utilizadores da aplicação. É responsável por guardar os dados da estadia dos clientes e está relacionada com outras 10 entidades. Esta é a tabela com mais relações em toda a aplicação. Assim sendo, podemos apelidá-la de entidade central da aplicação.

```
{ES_ID: int; CL_ID: int; ES_DATA_INICIO: date & time; ES_DATA_FIM: date & time;
ES_NUM_ACOMP: smallint; ES_NUM_QUARTOS: smallint; ES_PEDIDOS_ESP: varchar;
ES_ESTADO: int; ES_VALOR: decimal; ES_NUM_ADULTOS: smallint ; ES_NUM_CRIANCAS:
smallint; ES_NOITES: int ; ES_SALDO: decimal}
```


Produtos: Entidade responsável por guardar os dados dos produtos que são vendidos quando os clientes se encontram hospedados e pela gestão de *stocks*.

{PR_ID: int; PR_NOME: varchar; PR_MARCA: varchar; PR_DESCRICAO: varchar;
PR_REFERENCIA: varchar; PR_LOCAL: varchar; PR_TIPO: varchar; PR_QTD: int;
PR_PRECO: decimal; PR_DATA: date & time; PR_ESTADO: smallint}

Dados_Produtos: Um produto só pode ser vendido se existir uma estadia. Se o produto estivesse directamente relacionado com a entidade estadia, punha-se o problema de, ao alterar os dados do produto, como por exemplo, o preço, esta alteração afectar a estadia. Para resolver este problema foi criada a entidade Dados_Produtos. Esta entidade guarda os dados principais do produto na data da venda do mesmo. Assim, ao alterar os dados do produto, são garantidos os dados antigos como, por exemplo, o preço.

{ES_ID: int; PR_ID: int; DPR_PRECO: decimal; DPR_QTD: int; DPR_DATA: date & time}

Calendário: Entidade criada para guardar as datas das reservas/estadias.

{ID: int; ES_ID: int; TITLE: varchar; START: date & time; END: date & time;
BACKGROUND_COLOR: varchar}

Pagamentos: A todas as reservas está associado um ou mais pagamentos. A entidade pagamentos é responsável por guardar o valor do pagamento e a data em que foi efectuado.

{PG_ID: int; ES_ID: int; PG_DATA: date & time; PG_VALOR: decimal}

Tipo_Pag: Esta tabela serve para a normalização da base de dados. Assim, os dados ficam mais consistentes e com maior facilidade de acesso. Era possível escrever estes dados na tabela pagamentos mas, para optimizar, foi criada a tabela Tipo_Pag.

{TP_ID: int; PG_ID: int; TP_TIPO: varchar}

Dados_Estados: A estadia de um cliente passa por diferentes etapas, desde a pré-reserva até ao *check-out*. Esta entidade guarda os diferentes estados da estadia.

{DEST_ID: int; ES_ID: int; DEST_TIPO: int; DEST_DATA: date & time}

Home: Esta entidade guarda dados temporários que são apagados e escritos em função das estadias e dos quartos. Quando um cliente faz o seu *check-in* e passa a estar hospedado na pousada, são escritos os dados nesta tabela para serem mais facilmente visualizados pelo utilizador. Quando o cliente faz *check-out* os dados são apagados.

{HO_ID: int; QR_ID: int; ES_ID: int; HO_CLIENTE: varchar; HO_CHECK_IN: date; HO_CHECK_OUT: date; HO_ACOMP: varchar; HO_QR_CL: smallint; HO_ESTADO: smallint; HO_DESPESA: decimal}

Capacidade: Esta entidade guarda dados temporários para permitir que o algoritmo detecte se os quartos escolhidos na pré-reserva têm capacidade ou não para albergar os adultos e crianças escolhidas pelo utilizador.

{QR_ID: int; CAP_ADUL: int; CAP_CRI: int}

Disponibilidade: Todos os quartos passam por 3 estados: pré-reserva, ocupado ou disponível. A entidade Disponibilidade guarda os estados dos quartos e as suas datas. É esta a entidade responsável por mostrar se um quarto está disponível para aceitar uma reserva para as datas pretendidas.

{DIS_ID: int; QR_ID: int; DIS_ESTADO: smallint; DIS_DATA_INICIO: date & time; DIS_DATA_FIM: date & time; DIS_DATA: date & time}

Tipo Quartos: Em hotelaria é normal a descrição do tipo de quarto que o cliente aluga, se o quarto alberga 1, 2, 3 ou mais pessoas e que tipo de camas possui. Esta entidade foi criada para guardar estes dados.

{TQ_ID: int; QR_ID: int; TQ_TEXTO: varchar; TQ_DESCRICAO: varchar; TQ_DATA: date & time}

Imagens: Entidade que guarda as imagens dos quartos.

{IM_ID: int; QR_ID: int; IM_NOME: varchar; IM_ESTADO: int; IM_DATA: date & time; IM_PRINC: int}

Funcionários: Esta entidade é a quarta entidade mais importante da aplicação. Ela é responsável por guardar os dados dos funcionários e está relacionada com outras duas entidades.

{FN_ID: int; FN_NOME: varchar; FN_SOBRENOME: varchar; FN_EMAIL: varchar; FN_MORADA: varchar; FN_CC: varchar; FN_NIF: int; FN_NIB: varchar; FN_TIPO: varchar; FN_PASSWORD: varchar; FN_DATA_REG: date & time; FN_ACESSOS: int; FN_DATA_NASC: date; FN_TELEFONE: varchar; FN_ESTADO: smallint}

Notificações: Esta entidade guarda os dados das acções que os utilizadores fazem na aplicação. Por exemplo, se é feita uma reserva esses dados têm de ser guardados; quando foi feita e por quem. O propósito desta entidade é guardar esses dados.

{NT_ID: int; FN_ID: int; NT_LIDA: smallint; NT_TIPO: varchar; NT_TIPO_ID: int; NT_OBS: varchar; NT_DATA: date & time}

Pagamentos_Func: Todos os funcionários têm um vencimento. Esta entidade serve para guardar os dados desses vencimentos.

{PF_ID: int; FN_ID: int; PF_SAL_BRUTO: decimal; PF_SAL_LIQ: decimal; PF_BONUS: decimal; PF_PENALIZACAO: decimal; PF_DESCONTOS: decimal; PF_DATA: date & time; PF_OBS: varchar}

Chart: Esta entidade não está relacionada com nenhuma outra entidade da aplicação. Esta tabela foi criada para dados estatísticos e lida directamente com a API Zingchart.

{CH_ID: int; CH_NOME: varchar; CH_DADOS: int}

À semelhança da entidade Chart, mais 4 entidades foram criadas e que não estão relacionadas com nenhuma outra entidade na aplicação. São elas:

FO_Galeria: Esta entidade guarda principalmente os dados das imagens que irão aparecer no *Front-Office*.

{FG_ID: int; FG_NOME: varchar; FG_DATA: date & time; FG_ESTADO: int; FG_TIPO: varchar}

FO_Promoções: Esta entidade guarda principalmente os dados das promoções que irão aparecer no *Front-Office*.

{FP_ID: int; FP_TITULO_RED: varchar; FP_TITULO: varchar; FP_TEXTO: varchar; FG_DATA: date & time; FP_ESTADO: int; FP_IMAGEM: varchar}

FO_Serviços: Esta entidade guarda principalmente os dados dos serviços que irão aparecer no *Front-Office*.

{FS_ID: int; FS_TITULO_RED: varchar; FS_TITULO: varchar; FS_TEXTO: varchar; FS_DATA: date & time; FS_ESTADO: int; FS_IMAGEM: varchar}

Visitantes: A última entidade da aplicação é responsável por guardar os dados dos utilizadores que visitam a página web da pousada.

{LOG_ID: int; LOG_DATA: date & time; LOG_IP: int; LOG_HOST_NAME: varchar; LOG_PAIS: varchar; LOG_REGIAO: varchar; LOG_CIDADE: varchar; LOG_CONTINENTE: varchar; LOG_LAT: varchar; LOG_LONG: varchar}

Na Figura 9 são mostradas algumas das principais relações entre as entidades.

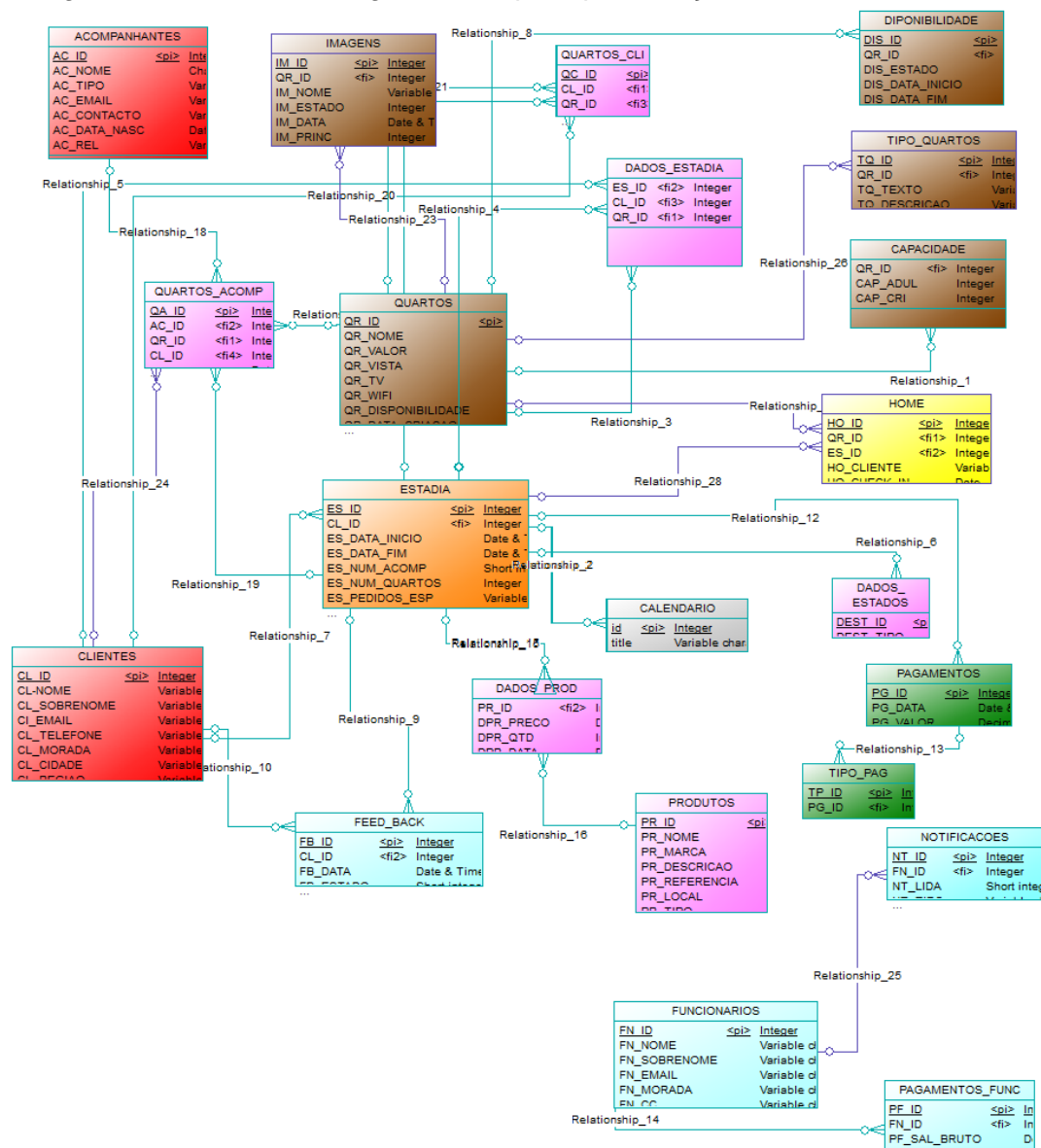


Figura 9 - Modelo ER Base de Dados

9.3. Anexo III - Manual do Utilizador

9.3.1. Reservas

9.3.1.1. Adicionar uma Pré-Reserva

Passo 1- No menu lateral, seleccionar a opção **Reservas** e escolher **Adicionar**.



Figura 10 - Adicionar Pré-Reserva - Passo 1

Passo 2- Escolher o cliente e carregar no botão **Próximo Passo**.

Nota 1- Se o cliente não existir na base de dados, terá de ser criado um. (**Ponto 9.3.6.1**).

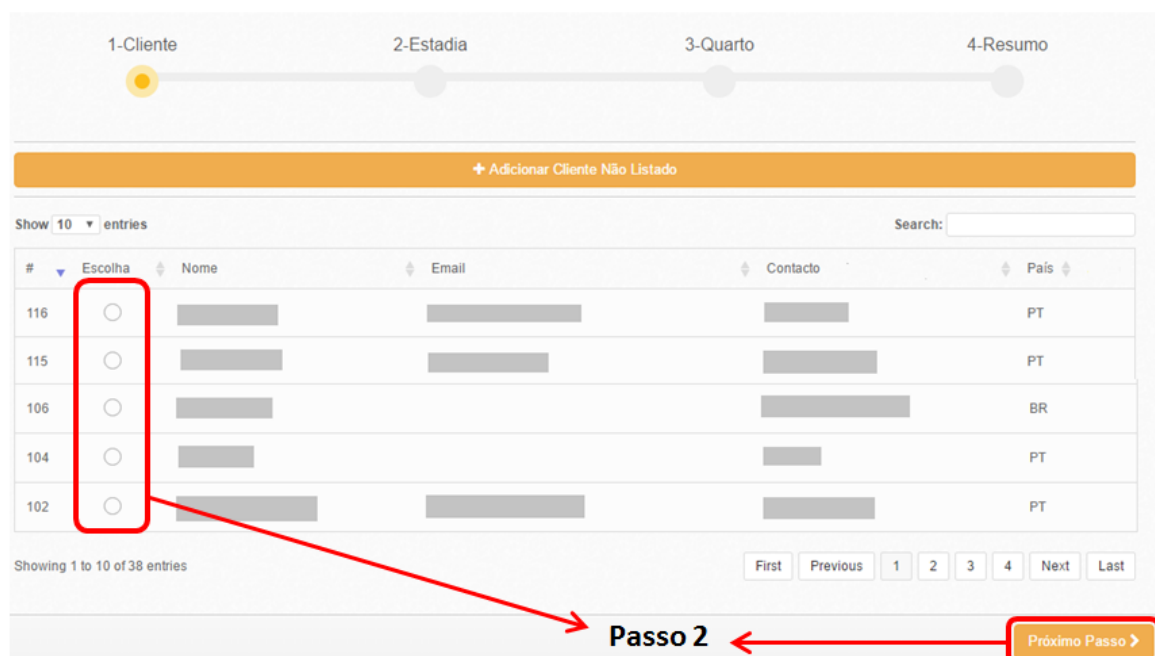


Figura 11 - Adicionar Pré-Reserva - Passo 2

Passo 3- Preencher correctamente os campos do formulário e carregar no botão **Próximo Passo**. Apenas o campo **Pedidos Especiais** não é de preenchimento obrigatório.

The screenshot shows the 'Passo 3' form with the following fields:

- Cliente: [Text input]
- Data Check-In: dd/mm/aaaa
- Data Check-Out: dd/mm/aaaa
- Número de adultos: 1
- Número de crianças: 0
- Transfer Aeroporto: ☐ € 5
- Transfer Cais das Velas: ☐ € 10
- Pedidos Especiais: [Text area]

At the bottom right, there are two buttons: 'Voltar' and 'Próximo Passo' (highlighted with a red box and arrow).

Figura 12 - Adicionar Pré-Reserva - Passo 3

Passo 4- Escolher o quarto onde o cliente irá pernoitar e carregar no botão **Próximo Passo**.

Nota 2- Os quartos que se encontram indisponíveis não são possíveis de seleccionar.

The screenshot shows the 'Passo 4' room selection screen with the following details:

- Informação:**
 - Adultos: 3
 - Crianças: 2
 - Pedidos Especiais:
- Terceira:**
 - Época: Alta, Normal, Baixa
 - Capacidade: 2 + 1
 - Tipo: Double
 - TV: ☒
 - WC: ☒
 - WIFI: ☒
 - VISTAS: ☒
 - Escolher: ☒
 - ☆ 75 €
- Flores:**
 - Época: Alta, Normal, Baixa
 - Capacidade: 2 + 0
 - Tipo: Twin
 - TV: ☒
 - WC: ☒
 - WIFI: ☒
 - VISTAS: ☒
 - Escolher: ☐
 - ☆ 75 €
- Faial:**
 - O Quarto já se encontra ocupado ou reservado para as datas pretendidas.
 - Capacidade: 2 + 0
 - Tipo: Twin
 - TV: ☒
 - WC: ☒
 - WIFI: ☒
 - VISTAS: ☒
 - ☆ 100 €

At the bottom right, there are two buttons: 'Voltar' and 'Próximo Passo' (highlighted with a red box and arrow).

Figura 13 - Adicionar Pré-Reserva - Passo 4

Passo 5- Após verificar se os dados estão inseridos correctamente, carregar no botão **Adicionar Pré-Reserva**. Uma janela modal aparece a confirmar a nova pré-reserva.

Passo 5

Quarto(s)	Cap.	Tipo	Noites	Preço por Noite	Época	Total
Terceira	2+1	Double	1	€ 90.00	€ -15.00 ☆	€ 75.00
Flores	2+0	Twin	1	€ 90.00	€ -15.00 ☆	€ 75.00
Total Final = Total x N° Noites + (Transfer)						€ 150.00

★Época Alta | ☆Época Normal | ☆Época Baixa

Voltar Adicionar Pré-Reserva

Figura 14 - Adicionar Pré-Reserva - Passo 5

9.3.1.2. Consultar uma Pré-Reserva (reserva por confirmar) ou uma Reserva Confirmada.

Passo 1- Nos botões do cabeçalho, carregar no botão **Reservas**.

Passo 2- Escolher qual o tipo de reserva que deseja consultar.

Passo 1

Passo 2

Reservas

Por Confirmar 3

Confirmadas 3

Todas as Reservas

Corvo Ocupado

S. Jorge Ocupado

Figura 15 - Consultar Pré-Reserva – Passos 1 e 2

9.3.1.3. Confirmar, Ver Informação ou Cancelar uma Pré-Reserva.

Passo 1- Carregar no botão **Reservas** para consultar as pré-reservas (Ver **Ponto 9.3.1.2**).

Passo 2- Escolher um botão do lado direito: Confirmar Pré-Reserva, Informação da Pré-Reserva ou Apagar Pré-Reserva.

Nota 1- Caso escolha o botão **Confirmar**, saltar para o **Passo 3**.

Nota 2- Caso escolha o botão informação **i**, uma página com a informação da pré-reserva é mostrada.

Nota 3- Caso escolha o botão cancelar **X**, saltar para o **Passo 4**.

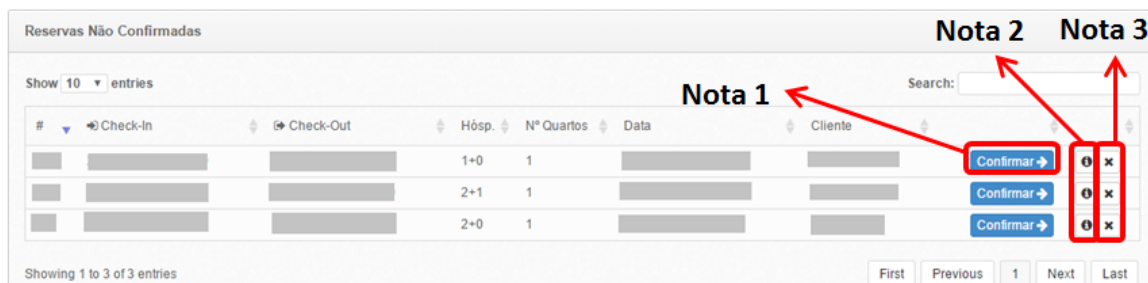



Figura 16 - Confirmar, Apagar ou Consultar Pré-Reserva

Passo 3- Após confirmar todos os dados, deve preencher o valor do sinal, seleccionar o método de pagamento e carregar no botão **Confirmar Reserva**. Uma janela modal aparece a confirmar a alteração da pré-reserva para reserva.

Figura 17 - Confirmar Pré-Reserva – Passo 3

Passo 4- Ao carregar no botão cancelar , uma janela modal aparece. Se deseja cancelar a pré-reserva, carregar no botão **Cancelar** se não, carregar no botão **Fechar**.

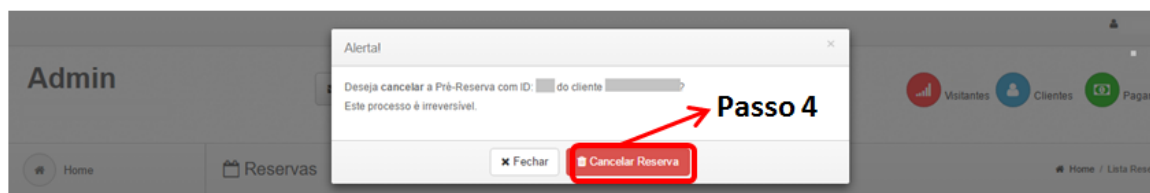




Figura 18 - Cancelar Pré-Reserva – Passo 4

9.3.1.4. Ver Informação ou Cancelar uma Reserva.

Passo 1- Carregar no botão **Reservas** para consultar as reservas (Ver **Ponto 9.3.1.2**).

Nota- Uma reserva só se encontra activa depois da confirmação de uma pré-reserva.

Passo 2- Escolher um botão do lado direito: informação da reserva  ou cancelar uma reserva . Os passos a seguir são iguais à **Nota 2** ou ao **Passo 4** do **Ponto 9.3.1.2**.

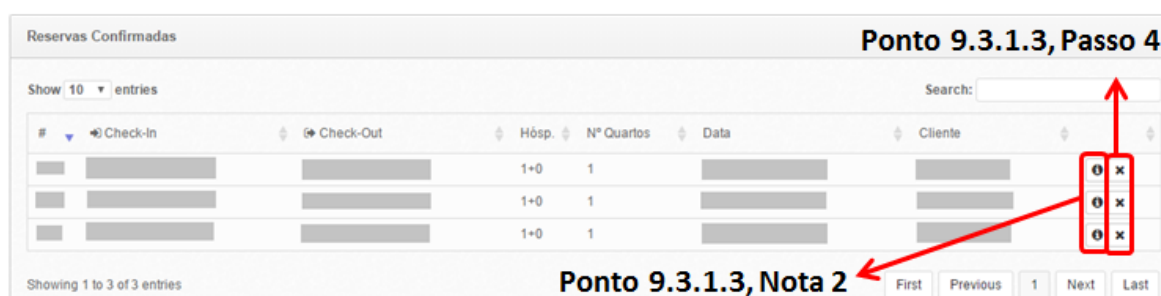


Figura 19 - Consultar ou Cancelar uma Reserva

9.3.2. Check-In

Nota 1- No botão **Check-In** (cabeçalho), a *label* verde mostra o número de *check-ins* pendentes para o dia.

Passo 1- Carregar no botão **Check-In** e carregar em **Ver Tudo** ou no nome do cliente.

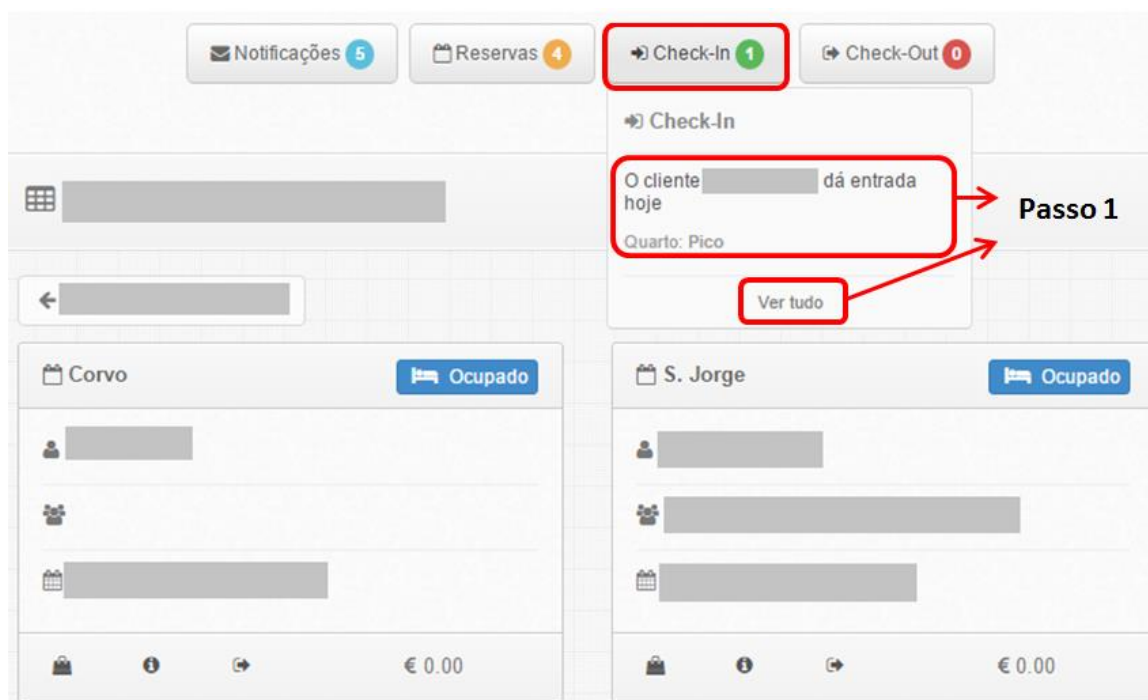


Figura 20 - Check-In - Passo 1

Passo 2- Na página **Check-In**, carregar no botão **Check-In**.

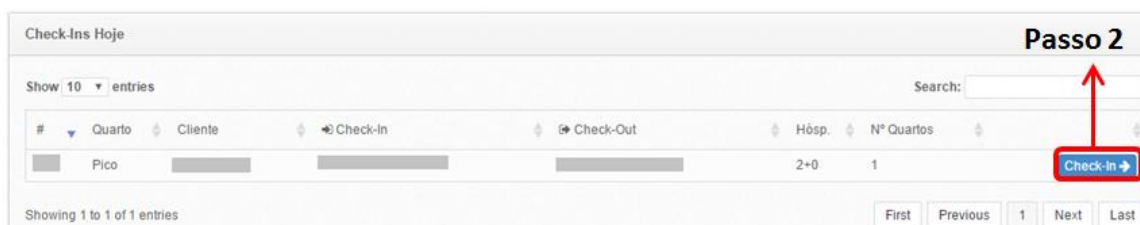


Figura 21 - Check-In - Passo 2

Passo 3- Preencher os dados do BI/CC e do NIF do Cliente.

Passo 4- Carregar no botão **Passo 2**.

1-Cliente 2-Acompanhante(s) 3-Quarto(s) 4-Pagamento

Cliente

BI/CC

NIF

Data Check-In

Data Check-Out

N° de Adultos 2

N° de Crianças 0

Pedidos Especiais

Passo 3

Passo 4

Voltar Passo 2 →

Figura 22 - Check-In - Passos 3 e 4

Passo 5- Preencher os dados de Acompanhantes (se existirem).

Passo 6- Carregar no botão **Passo Seguinte**.

1-Cliente 2-Acompanhante(s) 3-Quarto(s) 4-Pagamento

Acompanhante (1)

Nome Nome Sobrenome

Email lorenipsio@lorium.cmt

Tipo Adulto

Contacto (+000) 000000000

Data Nasc. dd/mm/aaaa

Passo 5

Passo 6

Voltar Passo Seguinte →

Figura 23 - Check-In - Passos 5 e 6

Passo 7- Distribuir o(s) hóspede(s) pelo(s) quarto(s).

Passo 8- Carregar no botão **Passo Seguinte**.

1-Cliente 2-Acompanhante(s) 3-Quarto(s) 4-Pagamento

Hóspedes

	Nome	Quarto
C		Terceira
A		
A		

Terceira

- ★ Época Alta
- ★ Época Normal
- ☆ Época Baixa

Capacidade: 2+1

Tipo: Double

TV: ☒ WC: ☒ WIFI: ☒ VISTAS: ☒

☆ € 75

Flores

- ★ Época Alta
- ★ Época Normal
- ☆ Época Baixa

Capacidade: 2+0

Tipo: Twin

TV: ☒ WC: ☒ WIFI: ☒ VISTAS: ☒

☆ € 75

Passo 7

Passo 8 Voltar Passo Seguinte

Figura 24 - Check-In - Passos 7 e 8

Passo 9- Verificar se todos os valores estão correctos e escolher o método de pagamento.

Passo 10- Carregar no botão **Confirmar Check-In**.

Nota 2- Após a confirmação de *check-in*, uma janela modal aparece a confirmar a acção.

1-Cliente 2-Acompanhante(s) 3-Quarto(s) 4-Pagamento

Contacto: Idade: Quarto: Terceira
 Contacto: Idade: Quarto: Flores
 Contacto: Idade: Quarto: Terceira

Check-In: Check-Out:
 N° de Noites: 1 Hóspedes: 2+1
 Transfer: Pedidos Especiais:

Quarto(s)	Cap.	Tipo	Noites	Preço por Noite	Época	Total
Terceira	2+1	Double	1	€ 90.00	€ -15.00 ☆	€ 75.00
Flores	2+0	Twin	1	€ 90.00	€ -15.00 ☆	€ 75.00
Total final = Total x N° Noites + (Transfer)						€ 150.00
Valor já pago						€ 75.00
Valor em falta						€ 75.00

Em falta € 75.00

Passo 9 Método Monetário

★Época Alta | ★Época Normal | ☆Época Baixa
 Voltar Confirmar Check-In

Passo 10

Figura 25 - Check-In - Passos 9 e 10

9.3.3. Check-Out

Nota 1- No botão **Check-Out** (cabeçalho), a *label* vermelha mostra o número de *check-outs* pendentes para o dia.

Passo 1- Carregar no botão **Check-Out** e depois em **Ver Tudo** ou no nome do cliente.

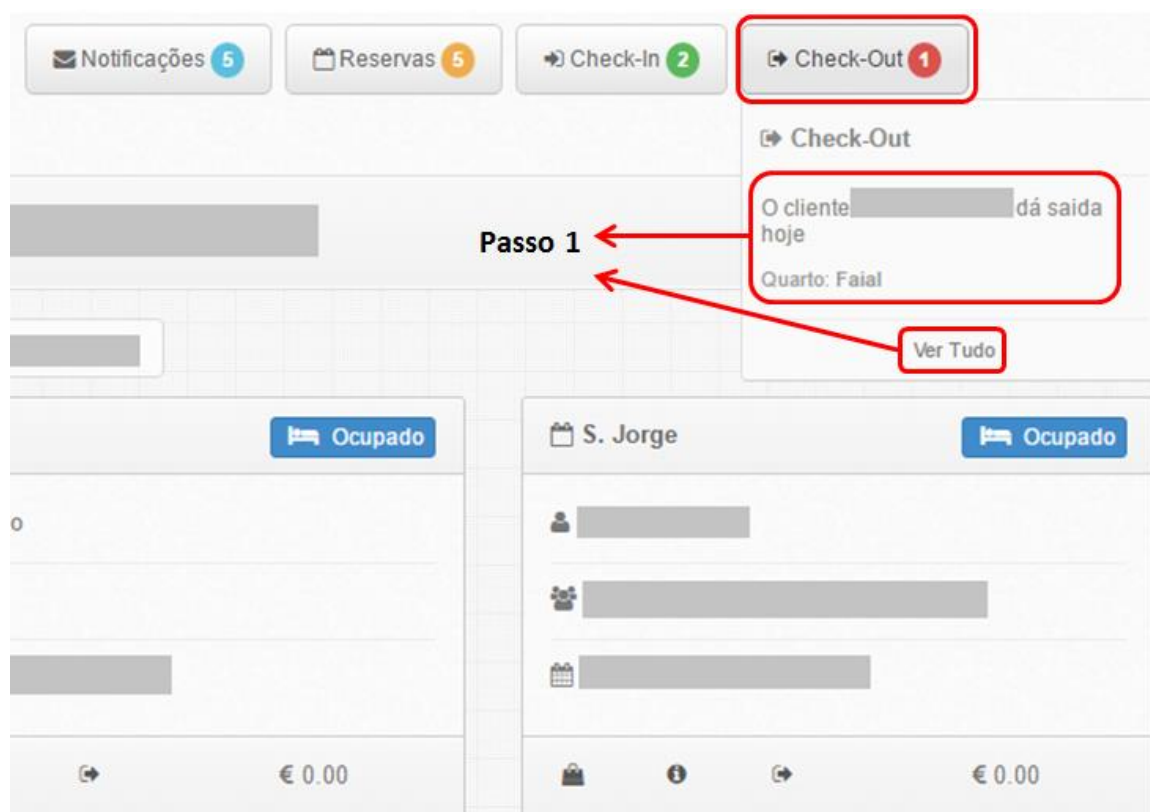


Figura 26 - Check-Out - Passo 1

Passo 2- Na página **Check-Out**, carregar no botão **Check-Out**.



Figura 27 - Check-Out - Passo 2

Passo 3- Verificar se todos os valores estão correctos, escolher método de pagamento e introduzir o valor em falta, se existir.

Passo 4- Carregar no botão **Confirmar Check-Out**.

Nota 2- Após a confirmação de *check-out*, uma janela modal aparece a confirmar a acção.

Cliente: [] Quarto: **Faial** Check-In: [] Transfer: Aeroporto: € 5.00
 Check-Out: [] Pedidos Especiais: []
 Nº de Noites: 1
 Hóspedes: 1+0

Quarto(s)	Cap.	Tipo	Noites	Preço por Noite	Época	Total
Faial	2+0	Twin	1	€ 115.00	€ -15.00 ☆	€ 100.00
Valor Final = Total x Nº de Noites + (Transfer)						€ 105.00
Valor já pago						€ 105.00

#	Produto(s)	Marca	Ref.	Qtd.	Preço (Unl.)	Sub-Total
Total da despesa						€ 0.00
Valor em falta						€ 0.00

★Época Alta | ☆Época Normal | ☆Época Baixa

Passo 3 → Valor: € 0.00
 Método: Monetário

Passo 4 → [Voltar] [Confirmar Check-Out ✓]

Figura 28 - Check-Out - Passos 3 e 4

9.3.4. Extras da Estadia

9.3.4.1. Adicionar/Remover Extras a uma estadia

Nota 1- Só é possível adicionar ou remover um extra a uma estadia se existirem clientes na pousada.

Passo 1- Na página principal (**Home**), carregar no ícone saco 🛍.

Corvo [Ocupado] S. Jorge [Ocupado] Terceira [Check-In]

Passo 1 → [🛍]

Figura 29 – Adicionar/Remover Extras - Passo 1

Passo 2- Na página **Extras**, escolher a quantidade de produtos. Quantidade positiva para adicionar ou quantidade negativa para remover produtos da estadia.

Passo 3- Carregar no botão **Adicionar**.

Nota 2- Após o produto adicionado o valor da estadia é alterado.

Produtos/Serviços

Cliente: [] Quarto: **Cunho** Check-In: [] Pedidos Especiais: []
 Acomp: [] S. Jorge Check-Out: []
 Acomp: [] S. Jorge N° de Noites: 1
 Hóspedes: 3+0

Passo 3

#	Produto(s)	Marca	Ref.	Qtd.	Preço (Unl.)	Sub-Total
2	Água Natural	Luso	43255786545	1	€ 3.25	€ 3.25
Total da despesa						€ 3.25

Show 10 entries

Passo 2

Produto	Marca	Descrição	Stock	Preço	Qtd.	
Água Natural	Luso	Água de nascente natural 20 ml	3	3.25	0	Adicionar
Água Natural	Dia	Água mineral natural 0,5 L	13	1.20	0	Adicionar
Vinho do Porto	Dona Antónia	Reserva Tawny 10 anos	9	35.68	0	Adicionar

Figura 30 – Adicionar/Remover Extras - Passos 2 e 3

9.3.5. Alterar Dados e Password da Conta

Passo 1- Carregar no botão superior direito que contém o nome do funcionário e escolher a opção **Conta**.

Passo 1

Notificações 5 Reservas 5 Check-In 2 Check-Out 0

Visitantes Cli Logout

Conta

Figura 31 - Alterar Dados e Password - Passo 1

Passo 2- Alterar os dados que deseja e preencher o campo **Password**.

Passo 3- Carregar no botão **Alterar**.

Nota- Após a alteração dos dados, um alerta aparece a confirmar.

The screenshot shows a form titled 'Alterar Dados e Password'. At the top, there are three tabs: 'Editar' (selected), 'Perfil', and 'Pagamentos'. The form is divided into two main sections. The first section, labeled 'Passo 2', contains fields for 'Contacto' and 'Morada'. The second section, labeled 'Passo 3', contains fields for 'Password', 'Nova password', and 'Conf. password'. A red box highlights the 'Password' section, and a red arrow points from the 'Alterar' button to the 'Passo 3' label.

Figura 32 - Alterar Dados e Password - Passos 2 e 3

9.3.6. Clientes

9.3.6.1. Adicionar Cliente

Passo 1- No menu lateral, escolher **Clientes**.

Passo 2- Escolher o botão **Adicionar Cliente**.

Nota 1- Do lado direito, ao carregar no botão informação **i**, uma página com a informação referente ao cliente é mostrada.

The screenshot shows the 'Clientes' page. On the left, there is a sidebar menu with options: 'Hóspedes', 'Reservas', 'Calendário', 'Produtos', 'Clientes' (selected), and 'Acompanhantes'. The main area displays a table of clients. The table has columns: '#', 'Nome', 'Email', 'Contacto', 'Registo', 'Estado', and 'Search'. The 'Estado' column shows 'Active' for all clients. A red box highlights the 'Clientes' button in the sidebar, labeled 'Passo 1'. Another red box highlights the 'Adicionar Cliente' button in the top right corner, labeled 'Passo 2'. A red arrow points from the 'Adicionar Cliente' button to the 'Passo 2' label. A red box also highlights the 'i' information icon in the 'Estado' column, labeled 'Nota 1'.

Figura 33 - Adicionar Cliente - Passos 1 e 2

Passo 3- Na página **Dados dos Clientes**, preencher os dados dos clientes.

Passo 4- Carregar no botão **Adicionar**.

Nota 2- Um alerta aparece com a confirmação do novo cliente.

The screenshot shows a form titled 'Dados dos Clientes' with the following fields: Nome, Sobrenome, Data de Nascimento (dd/mm/aaaa), Email (lorenipsio@lorium.cmt), Contacto (+(000) 000000000), Morada, Cidade, Distrito, Cód. Postal (ZIP/Postal), and Pais (Portugal). A red rounded rectangle encloses the entire form, with an arrow pointing to the text 'Passo 3'. At the bottom right, there are two buttons: 'Voltar' and 'Adicionar'. The 'Adicionar' button is highlighted with a red rectangle, and an arrow points from it to the text 'Passo 4'.

Figura 34 - Adicionar Cliente - Passos 3 e 4

9.3.6.2. Editar e Bloquear/Desbloquear Cliente

Passo 1- No menu lateral, escolher **Clientes**.

Passo 2- Do lado direito, carregar no botão editar ✎.

Nota 1- Para alterar os dados do cliente, seguir para o **Passo 3**. Para bloquear ou desbloquear um cliente, ir para o **Passo 4**.

The screenshot shows the 'Clientes' page. On the left, a sidebar menu has the 'Clientes' item highlighted with a red box and an arrow pointing to the text 'Passo 1'. The main area displays a table with columns: #, Nome, Email, Contacto, Registo, Estado, and actions. The 'Estado' column shows 'Activo' for all entries. The actions column contains edit (✎), delete (x), and add (+) icons. A red box highlights the action icons, with an arrow pointing to the text 'Passo 2'. The table has a search bar and a 'Show 10 entries' dropdown at the top.

Figura 35 - Editar e Bloquear/Desbloquear Cliente - Passos 1 e 2

Passo 3 – No formulário, alterar os dados que deseja.

Passo 4- Para bloquear um cliente, carregar no botão Estado **ON**. O estado passará para **OFF**. Caso o cliente esteja bloqueado e o pretenda desbloquear, carregar no botão Estado **OFF**, que passará a **ON**.

Passo 5 - Carregar no botão **Alterar**.

Nota 2 – Um alerta é mostrado a confirmar a alteração dos dados

The screenshot shows a form for editing a client. The form fields are: Nome, Sobrenome, Data de Nascimento, Email, Contacto, Morada, Cidade, Distrito, Cód. Postal, and País. At the bottom, there is an 'Estado' (Status) toggle switch currently set to 'ON' and an 'Alterar' (Edit) button. Red annotations indicate the following steps:

- Passo 3:** Points to the form fields, indicating where to enter or update client information.
- Passo 4:** Points to the 'Estado' toggle switch, indicating where to change the client's status.
- Passo 5:** Points to the 'Alterar' button, indicating where to click to save the changes.

Figura 36 - Editar e Bloquear/Desbloquear Cliente - Passos 3, 4 e 5

9.3.6.3. Remover Cliente

Passo 1- No menu lateral, escolher **Cientes**.

Passo 2- Do lado direito, carregar no botão apagar .

The screenshot shows the 'Clientes' (Clients) menu item selected in the sidebar. The main area displays a table of clients. Red annotations indicate the following steps:

- Passo 1:** Points to the 'Clientes' menu item in the sidebar.
- Passo 2:** Points to the 'Apagar' (Delete) button (represented by a red X icon) in the action column of the client table.

Figura 37 - Remover Cliente - Passos 1 e 2

Nota - O cliente só pode ser apagado se não tiver qualquer reserva associada a ele (seguir para **Passo 3**).

Caso contrário, uma janela modal aparece a alertar para este facto.

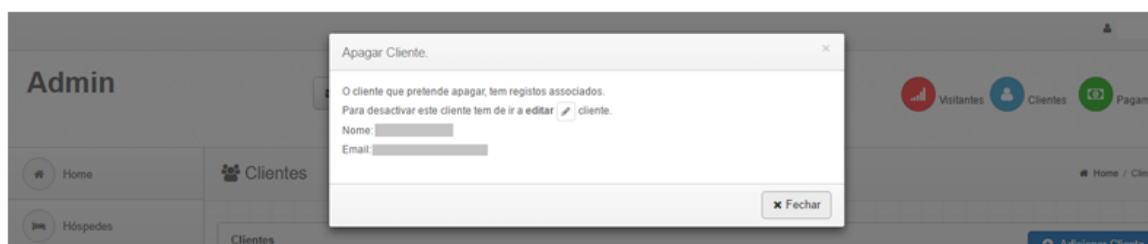


Figura 38 - Remover Cliente

Passo 3- Carregar no botão **Apagar** para eliminar o cliente.

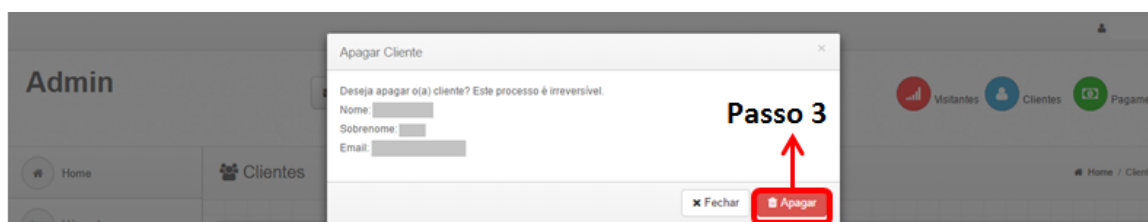


Figura 39 - Remover Cliente - Passo 3

9.3.7. Funcionários

9.3.7.1. Adicionar Funcionário

Passo 1- No menu lateral, seleccionar **Funcionários** e em seguida **Lista**.

Passo 2- Escolher o botão **Adicionar Funcionário**.


Nota 1- Do lado direito, ao carregar no botão informação , uma página com a informação referente ao funcionário é mostrada.



Figura 40 - Adicionar Funcionário - Passos 1 e 2

Passo 3- Na página **Adicionar Funcionário**, preencher os dados do funcionário.

Passo 4- Seleccionar o tipo de funcionário.

Passo 5- Carregar no botão **Adicionar**.

Nota 2- Um alerta irá aparecer com a confirmação do novo funcionário.

The screenshot shows a form for adding a new employee. The form fields are: Nome, Sobrenome, Data de Nascimento (dd/mm/aaaa), Email (lorenipsio@lorium.cmt), Contacto, Morada, BI/CC, NIF, and NIB (9999-9999-9999-9999-9999-99). At the bottom, there is a 'Tipo' dropdown menu set to 'Administrador'. To the right of the form, there are three annotations: 'Passo 3' with an arrow pointing to the 'Email' field, 'Passo 4' with an arrow pointing to the 'Tipo' dropdown, and 'Passo 5' with an arrow pointing to the 'Adicionar' button. The 'Adicionar' button is located at the bottom right of the form, next to a 'Voltar' button.

Figura 41 - Adicionar Funcionário - Passos 3, 4 e 5

9.3.7.2. Editar e Bloquear/Desbloquear Funcionário

Passo 1- No menu lateral, seleccionar **Funcionários** e em seguida **Lista**.

Passo 2- Do lado direito, carregar no botão editar ✎.

Nota 1- Para alterar os dados do funcionário, ir para o **Passo 3**. Para bloquear ou desbloquear um funcionário, seguir para o **Passo 4**.

The screenshot shows the 'Funcionários' (Employees) list in the PMS. The left sidebar contains a menu with 'Funcionários' and 'Lista' highlighted. The main area displays a table of employees with columns: Tipo, Nome, Email, Contacto, Registo, Acessos, Estado, and Actions. The 'Actions' column contains icons for edit (✎), delete (✖), and status (on/off). Annotations include 'Passo 1' pointing to the 'Lista' button in the sidebar and 'Passo 2' pointing to the edit icon in the 'Actions' column of the table.

Tipo	Nome	Email	Contacto	Registo	Acessos	Estado	Actions
1						Active	✎ ✖ ⚙
2						Active	✎ ✖ ⚙
3						Active	✎ ✖ ⚙
4						Active	✎ ✖ ⚙
5						Active	✎ ✖ ⚙

Figura 42 - Editar e Bloquear/Desbloquear Funcionário - Passos 1 e 2

Passo 3- No formulário, alterar os dados que deseja.

Passo 4- Para bloquear um funcionário, carregar no botão Estado **ON**. O estado passará para **OFF**. Caso o funcionário esteja bloqueado e o pretenda desbloquear, carregar no botão Estado **OFF**, que passará a **ON**.

Passo 5 - Carregar no botão **Alterar**.

Nota 2 – Um alerta é mostrado a confirmar a alteração dos dados.

The screenshot shows a form for editing an employee. A red rounded rectangle highlights the input fields for Nome, Sobrenome, Data de Nascimento, Email, Contacto, Morada, BICC, NIF, NIB, and Tipo (set to 'Funcionário'). A red arrow points from this area to the text 'Passo 3'. Below the inputs, the 'Estado' field shows a green 'ON' button, with a red arrow pointing to 'Passo 4'. At the bottom right, there are 'Voltar' and 'Alterar' buttons; a red arrow points from 'Alterar' to 'Passo 5'.

Figura 43 - Editar e Bloquear/Desbloquear Funcionário - Passos 3, 4 e 5

9.3.7.3. Remover Funcionário

Passo 1- No menu lateral, selecionar **Funcionários** e em seguida **Lista**.

Passo 2- Do lado direito, carregar no botão apagar .



Figura 44 - Remover Funcionário - Passos 1 e 2

Nota- O Funcionário só pode ser apagado se não tiver dados associados a ele. Uma janela modal aparece a alertar para este facto.

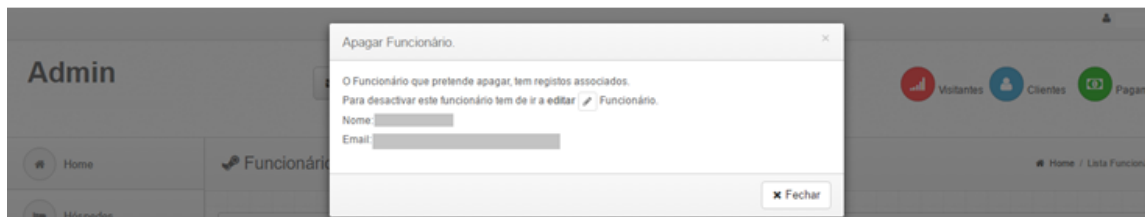


Figura 45 - Remover Funcionário

Passo 3- Carregar no botão **Apagar** para apagar o funcionário.

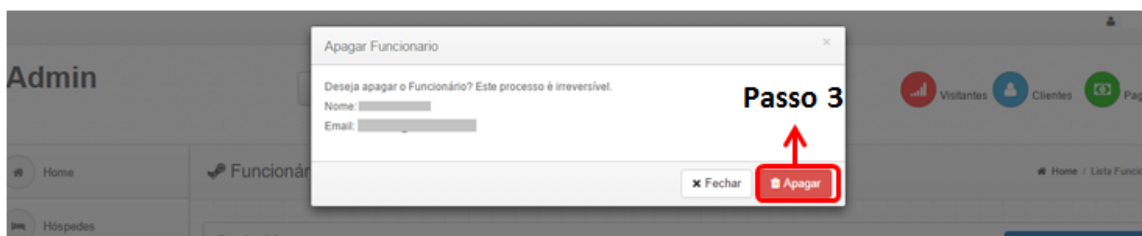


Figura 46 - Remover Funcionário - Passo 3

9.3.8. Vencimentos

9.3.8.1. Adicionar Vencimento

Passo 1- No menu lateral, seleccionar **Funcionários** e em seguida **Vencimentos**.

Passo 2- Escolher o botão **Adicionar Vencimento**.

Nota 1- Do lado direito, ao carregar no botão informação **i**, uma página com a informação referente ao vencimento do funcionário é mostrada.

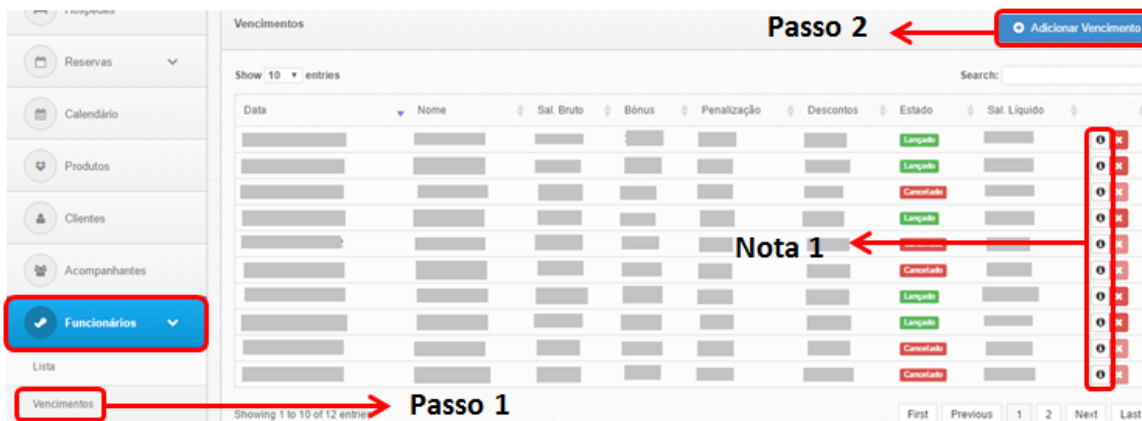


Figura 47 - Adicionar Vencimento - Passos 1 e 2

Passo 3- Seleccionar o funcionário relativo ao qual será adicionado o vencimento.

Nota 2- Caso o funcionário não exista na base de dados, carregue no botão **Novo Funcionário** para adicionar o novo funcionário. (Ver **Ponto 9.3.7.1, Passos 3, 4 e 5.**)

Passo 4- Preencher correctamente os campos do formulário.

Passo 5- Carregar no botão **Adicionar**.

Nota 3- Um alerta irá aparecer com a confirmação da criação de novo vencimento.

The screenshot shows the 'Adicionar Vencimento' form. A red box highlights the 'Funcionário' dropdown and the 'Novo Funcionário' button, with an arrow pointing to 'Passo 3'. Another red box highlights the 'Novo Funcionário' button, with an arrow pointing to 'Nota 2'. A red box highlights the 'Adicionar' button at the bottom right, with an arrow pointing to 'Passo 5'. The form fields include: Salário Bruto (€), Bónus (€), Penalização (€), Descontos (%), and Observações.

Figura 48 - Adicionar Vencimento - Passo 4

9.3.8.2. Cancelar Vencimento

Passo 1- No menu lateral, seleccionar **Funcionários** e em seguida **Vencimentos**.

Passo 2- Do lado direito, carregar no botão apagar .

The screenshot shows the 'Vencimentos' table. A red box highlights the 'Funcionários' button in the left sidebar, with an arrow pointing to 'Passo 1'. Another red box highlights the 'Vencimentos' button in the left sidebar, with an arrow pointing to 'Passo 1'. A red box highlights the 'X' delete buttons in the rightmost column of the table, with an arrow pointing to 'Passo 2'. The table has columns: Data, Nome, Sal. Bruto, Bónus, Penalização, Descontos, Estado, and Sal. Líquido. The 'Estado' column shows 'Liquido' and 'Cancelado'.

Figura 49 - Cancelar Vencimento - Passos 1 e 2

Nota- Uma janela modal aparece para confirmar o cancelamento do vencimento.

Passo 3- Carregar no botão **Cancelar** para eliminar o vencimento.

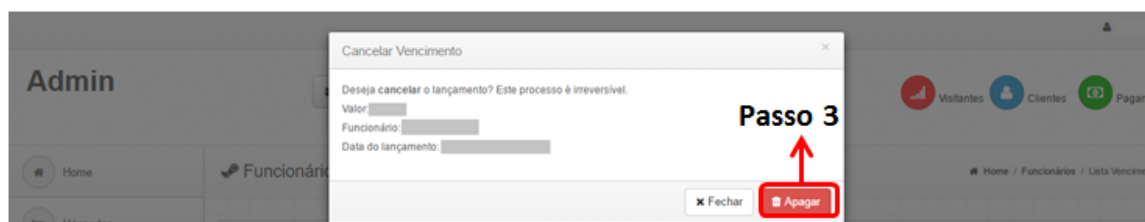


Figura 50 - Cancelar Vencimento - Passo 3